Trabajo de fin de Grado

Selección de características de audio para predecir el éxito de una canción mediante técnicas de aprendizaje estadístico

Francisco López Ayuso

Tutor: Emilio Parrado Hernández

Capítulo 1: Introducción

Motivación

A lo largo de las últimas décadas la música ha sufrido un proceso de globalización cada vez mayor. Debido a esto y al ímpetu humano de ordenar y categorizar cosas, desde esos comienzos de globalización se han creado los bien conocidos 'top 100', listas de canciones anuales que determinan qué canciones han sido las más escuchadas en todo el mundo en ese año, y por tanto las que más fama han adquirido. Por lo general, el género *Pop* es el que inunda todos los puestos en estas listas, de manera que los 'top 100' mundiales se consideran listas exclusivas de *Pop*. El uso de estas listas es muy habitual hoy día, y se ofrece a través de servicios web, revistas, programas de televisión, etc…

Estas listas 'top 100' sirven de referente para muchos fans de artistas *Pop* de fama mundial y oyentes en todo el mundo. Además de sus muchos seguidores, cualquier persona es capaz de reconocer este tipo de música como un producto de consumo, donde las melodías son fáciles, las formulas conocidas y el estilo similar. A pesar de ello, los éxitos *Pop* resultan pegadizos, y siguen inundando las listas 'top 100' mundiales.

De este hecho surge la cuestión a la que se trata de dar respuesta en esta propuesta: Si existe algún patrón detrás de las canciones que ocupan los primeros puestos de esas listas, y de ser así, encontrar los elementos que lo provocan.

El impacto de esta propuesta es grande debido a la gran cantidad de repositorios existentes y al hecho de que es posible acceder a ellos en general de forma sencilla y rápida. Asimismo, es importante destacar la facilidad con la que se puede intercambiar música en internet, ya que la escucha de música sobre soportes digitales de audio está altamente extendida y sigue hoy día en expansión.

Para la implementación de esta propuesta se ha decidido hacer uso de técnicas de aprendizaje estadístico, en este caso supervisado. Este tipo de técnicas se encuentran en auge y resultan muy útiles a la hora de generar modelos efectivos de predicción a partir de grandes cantidades de datos. Además permiten a cualquier persona extraer conocimientos varios y realizar diferentes estudios de bases de datos existentes. En esta propuesta se usan valores numéricos que representan diferentes características de una canción como datos brutos, sin embargo este tipo de técnicas se podrían aplicar de igual forma con otro tipo de datos, como por ejemplo imágenes, video, etc…

Descripción general del problema

La propuesta que aquí se desarrolla tratará de dar respuesta a la pregunta que se ha planteado anteriormente. Para comenzar se ha determinado un periodo temporal adecuado para llevar a cabo el análisis: 1980-2012. La idea inicial del estudio proponía analizar tan solo la década de los años 2000, sin embargo para considerar otros resultados posibles, se ha decidido finalmente abarcar esos 33 años. Además, los estilos musicales predominantes las décadas que comprenden esos años (’80, ’90 y ’00) son muy diferentes, lo que se ha considerado que podría resultar de interés para la propuesta.

Una vez establecido el periodo temporal, se ha elegido Billboard™ como medio para conocer las canciones que componen las listas de esas tres décadas. Billboard™ es una revista musical que ofrece entre otras noticias y servicios, listas ‘top 100’ desde hace años. La fama mundial de dicha revista y sus años de experiencia aportan una alta fiabilidad a la composición de sus listas ‘top 100’. El estudio a realizar busca encontrar si el éxito es predecible y a observar si existen características que estén más alineadas con las canciones exitosas o con las no exitosas. Además, observando dichas características y si el éxito resulta predecible, se podría definir el significado del éxito hasta cierto punto.

Si bien es cierto que el éxito de una canción puede estar determinado por muchos factores, fama del artista, vídeo musical, factores sociales, etc… y que además este está sometido a valoraciones subjetivas de todo tipo, para esta

propuesta se ha decidido centrarse en las canciones mismas, es decir, en el contenido de audio. Al ser el audio de una canción la información más directa que recibe el oyente, y al ser una componente común en todas las canciones que además es posible analizar de forma objetiva usando la estadística y el procesado se señal, se toma la componente principal del éxito.

Tomar nota de que se considera como audio las formas de onda de las canciones, las letras de estas entrarían en otros tipos de análisis de los cuales no se toma parte.

Para poder aplicar estadística a las canciones es necesario transformarlas en vectores observación con variables numéricas que recojan aspectos descriptivos de cada canción, pero que a su vez nos permitan comparar de modo fiable canciones de diversa índole y naturaleza. Por tanto, para realizar entonces un análisis de audio de todas las canciones que comprenden todas las listas de las 3 décadas y transformarlas en vectores observación, se ha hecho uso de los servicios de Echonest™, un servicio web que entre otras soluciones, ofrece una base de datos extensa de canciones sobre las que ha realizado un análisis de audio que resulta en 12 características que toman valores numéricos y que describen diferentes aspectos de la canción. Por tanto, de esta base de datos se han obtenido los datos brutos que se analizarán en esta propuesta.

Una vez obtenidos todos los datos brutos, es decir, todos los análisis de todas las listas de todos los años, se procede a realizar un análisis estadístico de los mismos, para poder extraer conclusiones. Se ha determinado que una canción exitosa es aquella que se encuentra entre los 20 primeros puestos de una lista, y se ha propuesto un análisis basado en la clasificación. Por tanto la base de los algoritmos de esta propuesta trata de clasificar las canciones entrantes en dos grupos, canción exitosa (20 primeros puestos) y canción no exitosa (resto de puestos), a partir de un aprendizaje supervisado con canciones de otras listas que sirven como datos de entrenamiento. Se han elegido tres métodos de clasificación, *KNN, Naive Bayes* y *SVM*. Estas tres tecnologías obedecen a principios de funcionamiento diferentes, de tal manera que será posible aislar en las conclusiones el efecto de una elección particular de un método de clasificación.

De esta manera se ha implementado la propuesta y se han obtenido diferentes resultados y conclusiones.

Objetivos

Habiendo establecido entonces la pregunta principal que servirá de motivación para esta propuesta y descrito de forma general el problema, se plantean los siguientes objetivos:

- Determinar a través de un método no causal de clasificación, si es posible predecir (clasificar de forma predictiva) si una canción será exitosa o no.

- Observar, utilizando el mismo método anterior, qué características se consideran las menos relevantes para realizar dichas predicciones.

- Analizar de forma independiente un periodo temporal que resulte de interés, o que aporte los mejores resultados, y buscar los factores o características que determinan el éxito en ese periodo.

- Determinar si existen interacciones entre características y ver cómo estas afectan a las clasificaciones realizadas.

Descripción de cada capítulo

A continuación se realiza una breve descripción del contenido de cada capítulo:

- **Capítulo 2: Estado del arte**

Se hará explicación de todos los servicios web y software utilizados, así como de los métodos matemáticos implementados en esta propuesta. También se incluye una descripción del marco legal bajo el que se encuentra la propuesta.

- **Capítulo 3: Diseño e implementación de la clasificación**

Se explicará en detalle cómo se ha realizado esta propuesta, describiendo con detalle todas las partes de las que se compone y su diseño.

- **Capítulo 4: Validación Experimental**

Se muestran los resultados obtenidos (gráficas, tablas, etc…) y se comenta brevemente su contenido al predecir el éxito de las canciones de Billboard™.

- **Capítulo 5: Conclusiones**

Se resumen las conclusiones a las que se ha llegado al haber realizado la propuesta a partir de los resultados obtenidos.

- **Capítulo 6: Presupuesto**

Se detalla el presupuesto que habría supuesto realizar esta propuesta en un entorno profesional.

Capítulo 2: Estado del Arte

En este capítulo se describirán las diferentes tecnologías y servicios utilizados para la realización de la propuesta. En primer lugar se hablará de los servicios web y el software de los que se ha hecho uso y en segundo lugar se describirán los métodos matemáticos que se han aplicado para el análisis de los datos recogidos.

Servicios web utilizados

 [1]

Billboard™ es una revista musical de fama mundial fundada en los Estados Unidos en 1984, que se publica de forma semanal. Trata noticias de actualidad en el mundo musical, abarcando diferentes géneros, como el *Pop, Rock, Gospel, Hip-Hop, RnB,* etc… Billboard™ cuenta además con su propia web www.billboard.com, donde se pueden acceder a las revistas ya publicadas a través de su hemeroteca digital, leer noticias cortas, escuchar listas de reproducción, ver videos musicales, ver su catálogo de actividades, consultar bibliografías de artistas, etc… Billboard™ es popular sobre todo debido a sus 'top 100', *Billboard* 's *Hot 100*™, listas anuales de los 100 mejores hits de ese año a escala mundial. En esta propuesta se desea analizar y determinar si es posible predecir el éxito musical, a través de 3 décadas de listas de éxitos musicales, con lo que los servicios que ofrece Billboard™ resultan extremadamente útiles y proporcionan una colección de datos brutos iniciales muy fiables.

[1] Fuente: www.billboard.com

A pesar de generar y contener estas listas en su web, el sistema de visualización que proponen resulta algo incómodo de manejar, ya que en cada puesto de cada lista no sólo se nombra el artista y título de la canción, sino que se muestra una imagen del artista así como una pequeña descripción de la canción. Es por ello que se han visualizado dichas listas a través de la web http://www.bobborst.com, un blog de un particular que contiene una base de datos de las listas 'top 100'que ha publicado Billboard™ desde 1946 hasta la actualidad. En este blog, las listas tan sólo contienen el nombre de la canción y el artista en cada puesto, lo que facilita enormemente la recopilación de datos numerosos como es el caso (parejas Artista-Título). La imagen 1 muestra el sistema de visualización que propone Billboard™, y en la imagen 2 muestra el sistema de visualización en el blog del particular.

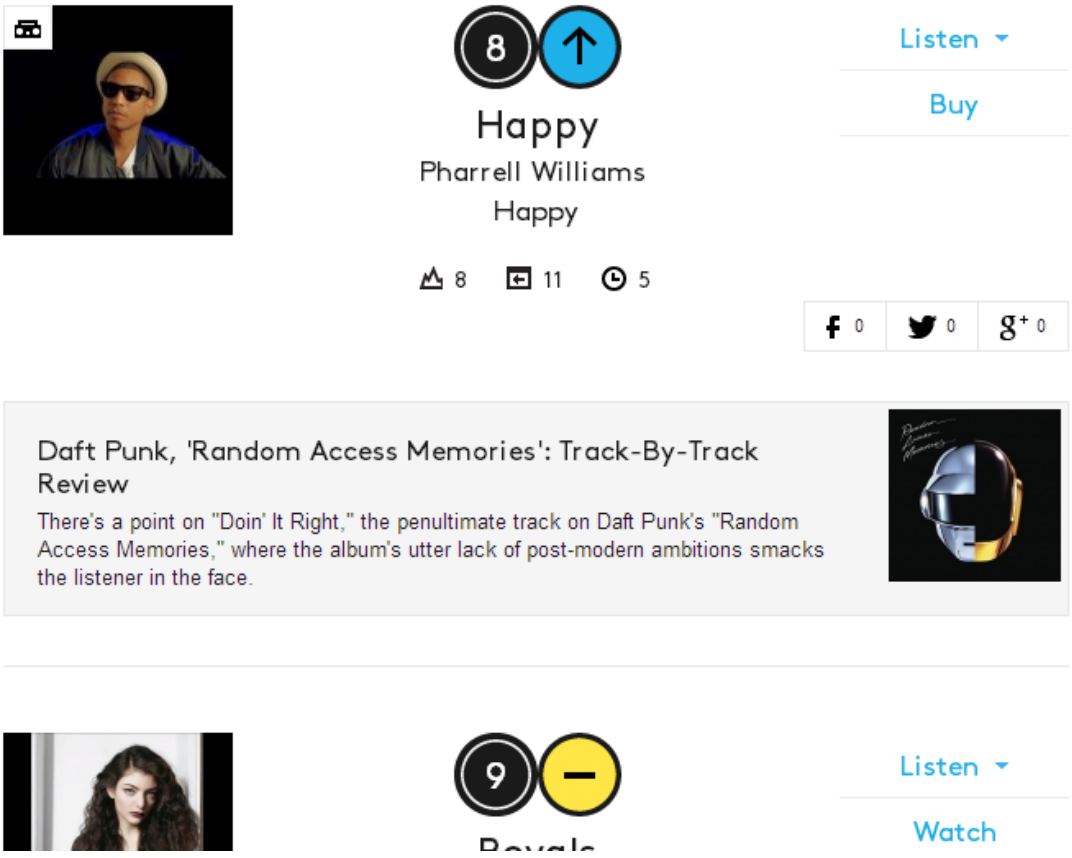


Imagen 1: Sistema de visualización de listas en www.billboard.com[2]

[2] Fuente: www.billboard.com

Play a countdown medley of the top 25 songs of the year Length: 1:53

You also might like the number one songs of 2012.

« 2011 **2012** 2013 »

Position	Artist	Song Title
1	Gotye feat. Kimbra	Somebody That I Used To Know
2	Carly Rae Jepsen	Call Me Maybe
3	fun. feat. Janelle Monae	We Are Young
4	Maroon 5 feat. Wiz Khalifa	Payphone
5	Ellie Goulding	Lights
6	The Wanted	Glad You Came
7	Kelly Clarkson	Stronger (What Doesn't Kill You)
8	Rihanna feat. Calvin Harris	We Found Love
9	Nicki Minaj	Starships
10	One Direction	What Makes You Beautiful
11	Flo Rida feat. Sia	Wild Ones
12	Adele	Set Fire To The Rain
13	LMFAO	Sexy And I Know It

Imagen 2: Sistema de visualización de listas en la web http://www.bobborst.com[3]

[4]

"Echonest™ es una compañía que ofrece servicios web de 'inteligencia musical', proporcionando a desarrolladores un conocimiento profundo de contenidos musicales y fans. Servicios musicales líderes (Clear Channel's iHeartradio™, MOG™, Rdio™, SiriusXM™, Spotify™), redes sociales multimedia (BBC.com™, Foursquare™, MTV™, Twitter™, VEVO™, Yahoo!™), manufacturadores de dispositivos conectados (doubleTwist™, Nokia™) y grandes marcas (Coca Cola™, Intel™, Microsoft™, Reebok™) usan su plataforma y servicios para construir experiencias musicales más inteligentes que ayudan a fans a

[3] Fuente: http://www.bobborst.com
[4] Fuente: www.echonest.com

descubrir, compartir e interactuar de una mejor manera con la música que les gusta."[5]

De entre los diferentes servicios y soluciones que propone Echonest™, en esta propuesta se ha hecho uso de las herramientas para la identificación de audio, un contenido que se ofrece de forma gratuita y con código abierto para usuarios registrados (el registro en la web también es gratuito). Echonest™ pone a disposición de dichos usuarios una API extensa que contiene todos los métodos y funciones, parámetros de entrada y de salida, requisitos, etc… que se implementan en sus servicios, así como ejemplos tanto de peticiones al servidor como las respuestas que se obtendrían. En conjunto con esta API se ponen también a disposición de forma gratuita a usuarios registrados diferentes librerías, tanto oficiales como no oficiales (creadas por usuarios), en varios lenguajes de programación, que incluyen *Phyton*™, *Java*™, *C++*™, *iOS*™ etc…

En esta propuesta se utiliza un sistema de búsqueda de canción que requiere conocer su identificador (una combinación de 18 letras y números que impone el propio servicio web y que es único de cada una), el nombre del artista y de la canción. Asimismo, para proceder a extraer los datos con los que se trabajarán y se realizarán los diferentes análisis que se proponen, se hace uso de la API y las librerías ofrecidas.[6] Los datos que se desean extraer de la base de datos de Echonest™ son 12 características con valores numéricos fruto de un análisis de audio realizado por el servicio web. Se pretende obtener estas 12 características para cada canción que figure en las listas que ofrece Billboard™, de tal manera que se acabe teniendo unas matrices de datos ordenados que corresponden al análisis de audio de dichas canciones.

El análisis de audio es realizado por la herramienta *Analyze*™. *Analyze*™ recibe un fichero digital de audio y genera una respuesta del tipo *JSON* que describe la estructura de la canción y su contenido musical, incluyendo ritmo, tono, y timbre. Toda la información proporcionada es precisa a la milésima de segundo.

[5] Texto: Traducción de www.echonest.com
[6] En el capítulo 2 se hace una revisión detallada de su uso

"Analyze™ hace uso de un algoritmo propio que simula cómo una persona escucharía música. Incorpora principios de psicoacústica, percepción musical, y un aprendizaje adaptativo que modela los procesos físicos y cognitivos de la escucha humana. Las respuestas que devuelve *Analyze*™ contienen una descripción completa de todos los eventos musicales, estructuras y atributos globales de la canción, como la clave (*Key*), volumen (*Loudness*), compás (*Time Signature*), tempo, ritmo (*Beats*), secciones y armonía.

De toda la gama de parámetros de salida que existen en *Analyze*™, para esta propuesta se han utilizado aquellos que procura *audio_summary,* un parámetro de entrada opcional en las peticiones a los servidores de Echonest™ que devuelve un grupo concreto de características[7]. Se clasificarían en dos grupos: Características de canción y Características Acústicas"[8].

- **Características de canción[9]**

  - *Time Signature* (compás): Es un valor estimado del compás de la canción. Este parámetro es una convención para determinar cuántos tiempos hay por compás, y se devuelve como un número natural.

  - *Key* (clave): Es un valor estimado de la clave de una canción. Este parámetro identifica la triada tónica, el acorde, mayor o menor, que representa la nota principal sobre la que se sostiene la canción. *Key* comienza por Do, y asciende en la escala cromática (por semitonos). Los valores que toma esta característica abarcan valores enteros desde 0 (Do) hasta 11 (Si).

  - *Mode* (modalidad): Indica la modalidad de una canción, mayor o menor, es decir, el tipo de escala de la que deriva su contenido melódico. Toma valores 0 (mayor) ó 1 (menor).

[7] Explicación detallada de peticiones y respuestas en Capítulo 2
[8] Traducción de [8]
[9] Características de Canción y Características Acústicas traducidas de www.echonest.com

- *Tempo:* Es el tempo estimado en general de una canción medido en BPM. En terminología musical, el tempo es la velocidad de una canción.

- *Loudness* (volumen): Es el volumen en general de una canción en decibelios. Los valores de volumen que maneja *Analyze*™ son promediados a lo largo de la canción y son útiles para para comparar niveles relativos en diferentes segmentos de la canción. El volumen es una cualidad del sonido que resulta en una correlación primaria psicológica de la fuerza física (amplitud).

- *Duration* (duración): La duración en segundos de una canción.

- **Características Acústicas**

Las características acústicas son estimaciones de cualidades subjetivas de una canción. Se modelan mediante aprendizaje y se devuelven como un número decimal que toma valores de 0.0 a 1.0.

- *Danceability* ('bailabilidad'): Describe como de útil resulta una canción para bailarla, usando un número de elementos musicales (cuanto más próximo el valor a 1.0, mejor resultará la canción para ser bailada). Entre los elementos musicales que mejor caracterizan la 'bailabilidad' se encuentran tempo, estabilidad en el ritmo, fuerza del ritmo y una regularidad en general.

- *Energy* (energía): Representa una medida perceptual de la intensidad y actividad que se generan en una canción. Canciones típicas que se clasificarían como energéticas serían rápidas, con mucho volumen y ruidosas. Por ejemplo, una canción del género *Death Metal* contiene

en general una gran energía, mientras que una pieza de *Bach* alcanzaría valores bajos de energía. Entre las características perceptuales que contribuirían a un valor alto de energía se encuentran el rango dinámico, volumen percibido, timbre y entropía general.

- *Speechiness*: Detecta la presencia de palabra hablada en una canción. Cuanta más cantidad de canción sea exclusivamente voz humana, más cerca de 1.0 se encontrará esta característica.

- *Liveness:* Detecta la presencia de público en la canción. Cuanto más probable sea que dicha canción se haya grabado en directo, más cerca de 1.0 estará su valor. Debido a la baja presencia de canciones grabadas en directo, el umbral para detectar *Liveness* es más bajo que el umbral que para detectar *Speechiness*.

- *Acousticness:* Representa la probabilidad de que la grabación haya sido creada a base de instrumentos acústicos y voz o por el contrario a base de instrumentos electrónicos, como por ejemplo instrumentos amplificados, sintetizados o con efectos. Canciones que hayan sido grabadas con guitarras eléctricas, sintetizadores, distorsión, voz tratada, etc… obtendrán valores bajos cercanos a 0.0, mientras que canciones grabadas con instrumentos de orquesta, guitarras acústicas o españolas, voz natural, baterías acústicas, etc… obtendrán valores cercanos a 1.0.

- *Valence:* Describe la positividad musical que proporciona una canción. Canciones con un valor alto de *Valence* sonarán más positivas, es decir, alegres, eufóricas, que animan, etc… mientras que canciones con un valor bajo sonarán más negativas, es decir, tristes, depresivas, etc… Esta característica en conjunto con *Energy* es un indicador fuerte del humor en una canción: las emociones en general que pueden caracterizar la acústica de una canción. Se debe tomar nota de que, en el caso de música con voz, esta pueda contener una

letra que difiera completamente del humor de la canción que se ha determinado.

Software utilizado

Para el desarrollo de esta propuesta se ha hecho uso de dos plataformas que implementan dos lenguajes de programación diferentes, Eclipse™ y MATLAB™.

- **Eclipse™**

Eclipse™ es una plataforma ofrecida de forma gratuita para la programación, compilación y creación de programas en Java™ y para el desarrollo de aplicaciones y herramientas[10]. La primera parte de esta propuesta está implementada en Java™ y hace uso de esta plataforma, de sus librerías incluidas por defecto y de librerías externas oficiales.

- *Librerías internas de Eclipse™ utilizadas*:
    - *java.io* y *java.nio*: Para el manejo de ficheros externos, incluyendo a los que se puedan generar desde Eclipse™.
    - *java.net*: Para realizar las peticiones *HTTP GET* necesarias para la extracción de datos de los servidores de Echonest™
- *Librerías oficiales externas utilizadas:*
    - *org.json:* Para el manejo de respuestas del servidor en fomato *JSON*. Esta librería permite la creación de objetos del tipo *JSON* a

[10] Para más información visitar https://www.eclipse.org

partir de respuestas en ese formato de un servidor, así como la posibilidad de acceder y manejar los datos que éstas contengan de manera sencilla y a través de los métodos contenidos en esta librería. *JSON* es un formato de intercambio de datos independiente de lenguajes de programación que representa dichos datos como objetos y arrays[11]. Se encuentra disponible de forma gratuita desde www.oracle.com.

- *org.apache.poi.hssf*: Para la creación de ficheros en formato *Excel*™ y su posterior manejo. Esta librería, *org.apache.poi,* permite el manejo de todo tipo de documentos que se puedan generar desde software de Microsoft™. Sin embargo se hace uso de la librería referente a archivos *Excel*™, *hssf*. Se ha decidido utilizar este tipo de ficheros ya que implementan de manera sencilla la acumulación de datos de forma ordenada a través de tablas. Además, la plataforma MATLAB™ (con la que se procesarán los datos recogidos) contiene una serie de métodos para su manejo que resultan muy intuitivos, rápidos y fáciles de implementar. Se encuentra disponible de forma gratuita desde http://poi.apache.org/.

- *com.echonest.api.v4*: Para el acceso a los diferentes métodos de los que hace uso Echonest™ para implementar sus servicios web. De esta librería sólo se hace uso de los métodos referentes a la subida de canciones a sus servidores y al análisis y obtención de características de las mismas[12]. Se encuentra disponible de forma gratuita desde www.echonest.com.

[11] Para más información visitar www.oracle.com
[12] Explicación detallada en Capítulo 3

- **MATLAB™**

MATLAB™ es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, la visualización y la programación. Mediante MATLAB™, es posible analizar datos, desarrollar algoritmos y crear modelos o aplicaciones. El lenguaje, las herramientas y las funciones matemáticas incorporadas permiten explorar diversos enfoques y llegar a una solución antes que con hojas de cálculo o lenguajes de programación tradicionales, como pueden ser C/C++ o Java™[13]. La utilización de MATLAB™ requiere la compra de una licencia, dependiendo su precio de los paquetes de herramientas adicionales que se deseen incluir, el tipo de usuario final, etc…

Se ha utilizado esta plataforma para el posterior análisis de los datos recogidos del servicio web de Echonest™. En esta propuesta se hace uso de tablas de datos extensas, modelos estadísticos de clasificación, manejo y cálculo de matrices y generación de gráficas y diagramas de barras, con lo que la plataforma MATLAB™ resulta altamente útil y eficaz. Al necesitar modelos estadísticos de clasificación y diferentes métodos estadísticos (media, desviación estándar, etc…) se ha hecho uso del *toolbox* (paquete de herramientas) *Statistics Toolbox*™, que se encuentra en el paquete básico de MATLAB™. Este paquete de herramientas proporciona algoritmos y herramientas para organizar, analizar y modelar datos multidimensionales.

Además de las diferentes opciones que se han mencionado, se ha hecho un uso extenso de la documentación que se encuentra en su web[14], ya que proporciona una explicación en detalle de cada función que se decida implementar, así como ejemplos, tutoriales varios y en la mayoría de los casos la teoría matemática asociada a dichas funciones o procesos. Esta documentación está disponible desde esta web de forma abierta.

[13] Texto obtenido de www.mathworks.com/products/matlab
[14] www.mathworks.com

Métodos matemáticos implementados

En esta propuesta se tratará de comprobar hasta qué punto el éxito en la música es predecible o esperable. A partir de los datos recogidos se desea hacer un análisis estadístico de los mismos, para poder obtener resultados que cumplan los objetivos propuestos. Para poder llevar esto a cabo se han decidido implementar diferentes modelos estadísticos de clasificación mediante métodos de aprendizaje supervisado, con el fin de poder llegar a predecir si un grupo de canciones nuevas se encontrarán entre los dos grupos en los que se pretende dividir los datos recogidos, canciones exitosas y canciones no exitosas[15]. A continuación se mostrarán los 3 tipos de clasificadores que se han implementado en esta propuesta. Recordar que se hace uso de estas 3 tecnologías para aislar el efecto que la elección de un clasificador concreto pudiese tener en los resultados.

- ### Clasificación por *KNN*

La clasificación por *KNN* (K-Nearest Neighbours, K-Vecinos más cercanos en castellano) es un método discriminativo no paramétrico, en el cual se tiene un conjunto de datos *X* con *n* puntos, y mediante una función de distancia concreta se buscan los *k* puntos más cercanos (vecinos) en el conjunto *X* dado un punto o un conjunto de puntos de búsqueda *Y,* y se utilizan para clasificar. Este clasificador es un clasificador que usa información eminentemente local dentro del espacio de entrada para realizar las clasificaciones.

"Se tiene un conjunto *X* con *N* casos, con *n* variables predictoras X_1,... X_n y una variable a predecir, la clase *C*. El algoritmo *KNN* predice que un nuevo punto perteneciente al conjunto *Y* pertenezca a la clase más común de entre los *k* vecinos más cercanos en el conjunto *X*. El término 'más cercano' es determinado

[15] Detalles de implementación en Capítulo 3

por una función de distancia, que comúnmente suele ser la norma del vector que separa a los dos puntos (Distancia Euclidea)"[16].

"En la siguiente imagen se muestra un pequeño ejemplo, donde se tienen N=21 muestras que pertenecen al conjunto X, n=2 variables predictoras, dos clases, k=3 y se evalúa la muestra entrante perteneciente al conjunto Y"[17].

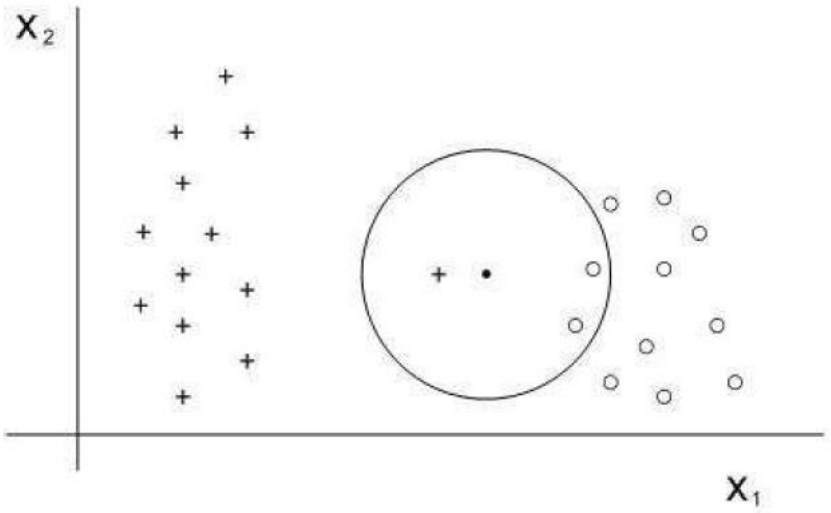


Imagen 3: Ejemplo de clasificación *KNN* con 3 vecinos y dos variables predictoras. Obtenida de [4]

En el ejemplo anterior se observa que al escoger como número de vecinos 3, se decide que la clase a la que pertenecerá la muestra del conjunto Y será aquella a la pertenezcan las muestras del conjunto X representadas en la imagen por círculos.

- **Clasificación con *SVM***

La clasificación con *SVM* (Support Vector Machines, Máquinas de vectores de soporte en castellano) es un método de clasificación discriminativo y semiparamétrico no lineal que estima una función de decisión lineal con la

[16] Traducido de [3]
[17] Traducido de [4]

particularidad de que se necesita un mapeo previo de los datos en un espacio de dimensiones más altas. Este mapeo está caracterizado por la elección de un conjunto de funciones conocidas como núcleos (en inglés, *kernels).*

Para implementar la clasificación por *SVM,* se aplica un acercamiento discriminativo, es decir, "se selecciona una regla de clasificación paramétrica (también denominada función discriminativa f_w(x) la cual determina la clase de una muestra *x,* y usa los datos para dar valores a los parámetros *w"*[18].

A la hora de considerar problemas de clasificación con *SVM* se pueden diferenciar dos tipos, aquellos con una geometría separable de forma lineal y aquellos con una geometría no separable de forma lineal.

- *Problemas con una geometría separable de forma lineal*

"El entrenamiento de un clasificador para un problema separable de forma lineal consiste en determinar el valor de su vector de pesos *W* y el término de sesgo *b*. Normalmente involucra la optimización de una funcionalidad que incluye un término de penalización favoreciendo un número reducido de errores entre los datos de entrenamiento, más algunos términos de regularización que aseguran una buena capacidad de generalización (buenas tasas de clasificación con datos que no se han usado para entrenar el clasificador).

En el caso concreto de *SVM*, "se intenta encontrar el híperplano separador óptimo, definido como aquél que maximiza el margen de clasificación (Ver Imagen 4). Este margen se calcula como la distancia entre la frontera de clasificación y la muestra más cercana de cada clase. Nótese que al generar una clasificación lineal con *SVM* no sólo se obtiene un error nulo en el conjunto de entrenamiento, sino que el margen que se implementa desde la frontera de clasificación presenta unos riesgos bajos si se diera una mala clasificación de las muestras de entrenamiento"[19]

$$J_{SVM}(w, b) = \frac{1}{2}||w||^2 \quad [20]$$

[18] Traducido de [7]
[19] Traducido de [7]
[20] Obtenido de [7]

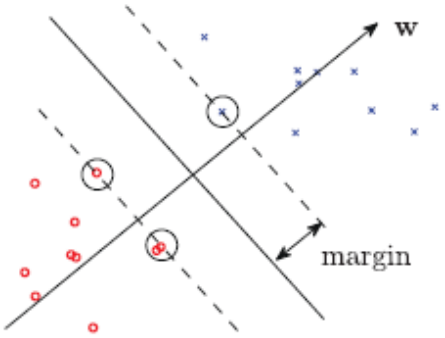


Imagen 4: Clasificación lineal utilizando *SVM*. Los vectores de soporte que definen *w* están rodeados con un círculo. Estas muestras recaen en el margen. Obtenida de [7]

- *Problemas con una geometría no separable de forma lineal*

"Para problemas con una geometría no separable de forma lineal se hace uso del concepto de margen suave, que consiste en añadir un conjunto de variables estacionarias no negativas que permiten la introducción de errores en el conjunto de entrenamiento. Por tanto se define la funcionalidad para el caso no separable como:

$$min_{w,b,\varepsilon}\left[\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\varepsilon_i\right]$$ [21]

Donde C es un híper parámetro que aplica un compromiso entre la maximización del margen y la minimización de los errores al entrenar.

Por otra parte, las *SVM* se pueden extender a una versión no lineal a través de lo que se conoce como *kernel-trick*. Un *kernel-trick* es una manera directa de derivar versiones no lineales de algoritmos donde los datos de entrada aparecen exclusivamente en productos escalares. Esta manera de proceder consiste en reemplazar todos los productos escalares entre los patrones de entrada por evaluaciones de una función *kernel*. Este procedimiento asegura que los datos

[21] Obtenido de [7]

que son mapeados de forma no lineal a un espacio de características donde el producto escalar entre dos ejemplos mapeados se pueda realizar evaluando una función *kernel* en el espacio de entrada."[22]

- **Clasificador *Naive Bayes***

"La clasificación por *Naive Bayes* es un método de clasificación que se basa en un modelado generativo probabilístico. En general, los clasificadores bayesianos clasifican con la probabilidad a posteriori. El aprendizaje en este tipo de clasificadores se puede simplificar en gran manera asumiendo que las características son independientes dada una clase, esto es $P(X|C) = \prod_{i=1}^{n} P(X_i|C)$, donde $X = (X_1,\dots X_n)$ es un vector de características y C es una clase"[23]. La clase predicha se tiene con:

$$\hat{C} = argmax_C P(C|X)$$

"Si se desea realizar una clasificación a partir de los datos de entrenamiento X, podemos minimizar el error realizando $argmax_y P(X|Y)$, donde Y pertenece al conjunto de clases C_k con k clases. Se busca estimar $\hat{P}(X|Y)$ a partir de $P(X|Y)$ y realizar la clasificación seleccionando $argmax_y \hat{P}(X|Y)$. De la definición de probabilidad condicional se tiene:

$$P(X|Y) = P(y,x)/P(x)$$

Por tanto se concluye que $argmax_y P(X|Y) = argmax_y P(y,x)$. Aplicando la regla del producto se tiene $P(y,x) = P(y)P(x|y)$. Si se asume que todas las características son independientes de la clase, entonces:

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|y)$$

Por tanto el clasificador *Naive Bayes* clasifica seleccionando:

[22] Traducido de [7]
[23] Traducido de [5]

$$argmax_y \, \hat{P}(Y) \prod_{i=1}^{n} \hat{P}(x_i|y))$$

Donde $\hat{P}(Y)$ y $\hat{P}(x_i|y)$ son estimaciones de las probabilidades respectivas que derivan de la frecuencia de sus características respectivas en el conjunto de muestras de entrenamiento"[24].

"A pesar de estas asunciones poco realistas el clasificador *Naive Bayes* que resulta funciona bien en la práctica, a menudo compitiendo con otras técnicas más sofisticadas"[25].

Marco Legal

Sobre las APIs y servicios ofrecidos por Echonest™ y los datos que estos proporcionan, se puede encontrar una explicación detallada de las condiciones de servicio y términos legales en su web.[26] Se debe comentar que, para la obtención de una *API KEY*[27], el usuario de Echonest™ una vez registrado, debe realizar la petición de dicha clave dando a conocer el uso que dará a los servicios de Echonest™. Sólo es posible hacer uso de los servicios de Echonest™ si se aprueba tal petición.

Respecto a Europa y España, el marco legal bajo el que se realiza la propuesta consta de las siguientes leyes:

- *Directiva 2006/116/EC del Parlamento Europeo y del Consejo, de 12 de Diciembre sobre las condiciones de la protección de copyright y de derechos relacionados concretos*

- *Directiva 2009/24/EC del Parlamento Europeo y del Consejo, de 23 de Abril sobre la protección jurídica de programas de ordenador*

[24] Traducido de [6]
[25] Traducido de [5]
[26] http://developer.echonest.com/terms
Y en el apartado *'Ground Rules'* en http://developer.echonest.com/docs/v4

[27] Detalles en Capítulos 2 y 3

- *Directiva 96/9/EC del Parlamento Europeo y del Consejo, de 11 de Marzo sobre la protección jurídica de bases de datos*

- *Real Decreto Legislativo 1/1996, de 12 de Abril, por el que prueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.*

- *Ley 5/1998, de 6 de marzo, de incorporación al Derecho español de la Directiva 96/9/CE, del Parlamento Europeo y del Consejo, de 11 de marzo de 1996, sobre la protección jurídica de las bases de datos.*

- *Ley 16/1993, de 23 de diciembre de incorporación al Derecho español de la Directiva 91/250/CEE, de 14 de mayo de 1991, sobre la protección jurídica de programas de ordenador.*

Capítulo 3: Diseño e implementación de la clasificación

La solución propuesta para llevar a cabo los objetivos descritos en la introducción se puede dividir en dos partes. Cada una está bien diferenciada, están implementadas de forma individual y utilizan diferentes tecnologías. Estas partes son:

- Obtención de las características de las canciones que figuran en las listas de éxitos de Billboard™ a través de los servicios que ofrece Echonest™
- Análisis estadístico del conjunto que forman los datos recogidos

En la primera parte de esta propuesta se generan 33 listas de 50 canciones cada una, con 12 características que corresponden a un análisis de audio de las mismas. Esas 33 listas corresponden a los años que transcurren entre 1980 y 2012. A pesar de que Billboard™ ofrece listas de 100 canciones, se han elegido las 50 primeras de cada lista para que formen las listas que aquí se utilizan. Esto ha sido así para simplificar el coste computacional al ejecutar el análisis estadístico posterior. En la segunda parte se procede al análisis estadístico de dichas características y a la obtención de los resultados.

Obtención de Características

Echonest™ contiene una gran base de datos de canciones, que contiene información relativa a ellas (artista, título, álbum, año, listas de reproducción, biografías de artistas etc…) así como un análisis de audio de las mismas. En esta propuesta se ha implementado un código en Java™ con la ayuda de la plataforma Eclipse™ que devuelve los datos brutos que se extraen de Echonest™ en forma de hoja de *Excel* que sirven para realizar el análisis matemático que se propone: listas de diferentes años con 50 canciones con 12

características cada una, fruto del análisis que realiza Echonest™ de cada una ellas.

Para obtener las características de una canción es necesario en primer lugar conocer el identificador de la misma, una combinación de 18 letras y números que impone el propio servicio web y que es único de cada una (Ej: *RadioHead – Karma Police:* SOCZZBT12A6310F251).

Echonest™ ofrece sus servicios y librerías a usuarios registrados de forma gratuita, y para ello otorgan a estos una clave de 17 caracteres, *API KEY,* que se debe incluir tanto en las peticiones *HTTP-GET* como en el acceso a las librerías (Ej: FILDTEOIK2HBORODV).

El conjunto de métodos para la obtención de parámetros y características que ofrece Echonest™ es aplicable de dos maneras: Mediante peticiones *HTTP-GET* que devuelven respuestas del tipo *XML/JSON* o mediante librerías que se encuentran disponibles para diferentes lenguajes de programación desde su web.

En esta propuesta se hace uso de cada una dependiendo únicamente de si la canción de la que se pretende obtener información (identificador y características) se encuentra en la base de datos de Echonest™ o no.

- **Canciones en la base de datos de Echonest™: Métodos con *HTTP-GET***

Antes de describir cómo se ha implementado la extracción de características mediante estos métodos, es necesario comentar que todas las respuestas que se reciben de los servidores de Echonest™ al realizar una petición *HTTP-GET* pueden ser del tipo *XML* o *JSON*. En el código implementado en Java™ se hace uso de las respuestas del tipo *JSON*, debido a que su manejo ha resultado ser más sencillo.

Si la canción se encuentra en la base de datos de Echonest™, se procede a conocer su identificador mediante un método que realiza una petición *HTTP-*

GET haciendo uso de las funciones que se encuentran disponibles en las librerías oficiales para Java™, que recibe el nombre de la canción, artista y la *API KEY* como parámetros de entrada, y que devuelve como salida el identificador de dicha canción en forma de "*String*". Este método crea una *URL* a partir de sus parámetros de entrada, realiza la petición *HTTP-GET* y recibe internamente una respuesta *JSON*, de la que se extrae el identificador de la canción utilizando métodos de las librerías oficiales de *JSON*. El ejemplo 1 muestra una *URL* completa para la obtención del identificador de la canción *Karma Police* de *RadioHead*. Además de los parámetros de entrada que recibe el método, se incluyen en la *URL* la especificación del formato de la respuesta que se recibirá, *JSON*, y el número de respuestas que se desean recibir, en todos los casos 1 (todas las peticiones son individuales, es decir, cada respuestas se recibe por separado).

http://developer.echonest.com/api/v4/song/search?api_key=FILDTEOIK2HBORODV&format=json&results=1&artist=radiohead&title=karma%20police

Ejemplo 1: URL para la obtención del identificador de una canción[28]

A continuación se muestra el ejemplo 2, la respuesta *JSON* que se obtendría al ejecutar la petición *HTTP-GET* anterior. La respuesta se muestra de forma ordenada para facilitar la vista de su contenido:


```
{
        "response": {
                "status": {
                        "code": 0,
                         "message": "Success",
                          "version": "4.2"
  },
                "songs": [
                        {
```


[28] Fuente: www.echonest.com

```
                    "artist_id": "ARH6W4X1187B99274F",
                    "id": "SOCZZBT12A6310F251",
                    "artist_name": "Radiohead",
                    "title": "Karma Police"
                    }
                ]
        }
}
```


Ejemplo 2: Respuesta *JSON* para la búsqueda del identificador de una canción[29]

Una vez obtenido el identificador de la canción, se utiliza un método que recibe como parámetros de entrada dicho identificador, la *API KEY* y un objeto *JSON*, que servirá para almacenar la respuesta del servidor, es decir todas las características de la canción, y que además se utilizará como parámetro de salida para su posterior utilización. Se crea una *URL* que contiene el identificador de canción, y la respuesta *JSON* con todas las características de la canción se devuelve como salida. En el ejemplo 2 se muestra la formación de una *URL* para la obtención de características de la canción *Stay Fly* de *Wil-Lean*. La petición contiene además de los parámetros de entrada del método en el que se realiza, la especificación del formato de la respuesta que se recibirá, *JSON*, el número de respuestas que se desean recibir (sólo 1), y lo que se desea recibir como respuesta, *audio_summary*, un parámetro que devuelve las características de la canción.


```
http://developer.echonest.com/api/v4/song/profile?api_key=FILDTEOIK2HBORODV&format=json&results=1&id=SOCZMFK12AC468668F&bucket=audio_summary
```


Ejemplo 3: URL para la obtención de las características de una canción[30]

El tipo de respuesta *JSON* que se obtendría es similar a la anterior. A continuación se muestra el ejemplo 4, lo que se obtendría al ejecutar la petición previa. La respuesta se muestra de forma ordenada para facilitar la vista de su contenido:

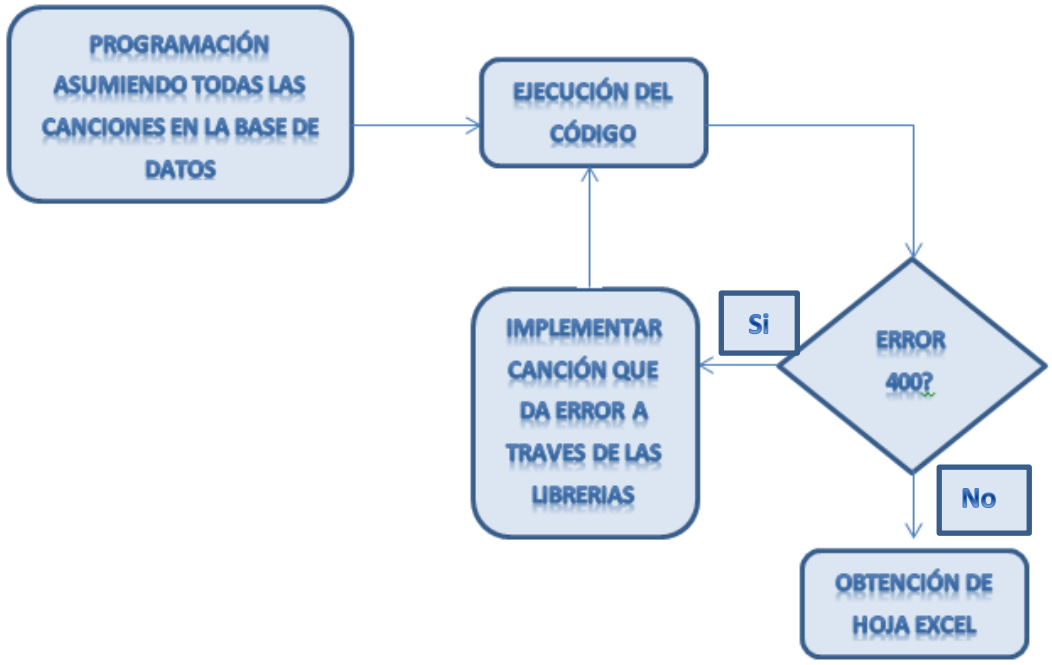
[29] Fuente: www.echonest.com
[30] Fuente: www.echonest.com

```
{
        "response": {
                "status": {
                        "version": "4.2",
                        "code": 0,
                        "message":
                        "Success"},
        "songs": [
                {"artist_id": "ARZHQSP12FE086C216",
                "artist_name": "Wil-Lean",
                "id": "SOCZMFK12AC468668F",
                "audio_summary": {
                        "key": 11, "analysis_url": "http://echonest-
analysis.s3.amazonaws.com/TR/yYGwoAXV4XSZe-
d6ZSJONlRvDvMog5S7z3EGCSVUfjnUQ7sac%3D/3/full.json?AWSAccessKeyId=AKIAJRDFE
Y23UEVW42BQ&Expires=1391597762&Signature=w44u5fsItc4CCSARdU4YHultcCE%3D",
                "energy": 0.40958800000000001,
                "liveness": 0.119308,
                "tempo": 147.85400000000001,
                "speechiness": 0.27377899999999999,
                "acousticness": 0.29020200000000002,
                "mode": 0,
                "time_signature": 4,
                "duration": 207.62076999999999,
                "loudness": -8.8699999999999992,
                "audio_md5": "a39ccbc9dc71df6fe9b62637f0cee49e",
                "valence": 0.33200099999999999,
                "danceability": 0.81364599999999998
                },
                "title": "Stay Fly"
                        }
                ]
        }
}
```


Ejemplo 4: Respuesta *JSON* para la obtención de características de una canción[31]

[31] Fuente: www.echonest.com

Para determinar si una canción existe o no en la base de datos de Echonest™, se escribe el código correspondiente a una lista de Billboard™, presuponiendo que todas las canciones que figuran en ella se encuentran en su base de datos, y se ejecuta. A partir de aquí se debe observar si el código genera alguna respuesta de error (todo el mismo previamente depurado y compilado). Entonces las únicas respuestas de error que se pueden generar son respuestas de los servidores de Echonest™, que se utilizan para determinar problemas con ciertas peticiones. Más concretamente, si una de las peticiones devuelve un código de respuesta *HTTP* 400 (*Bad Request*), la canción no existe en su base de datos, y por tanto será necesario obtenerla en formato digital y procesarla con las librerías que ofrece Echonest™ para Java™. Aquí se muestra un diagrama de flujo que lo ilustra:



- **Canciones fuera de la base de datos Echonest™: Métodos con las librerías para Java™**

Si existe una canción que no se encuentra en la base de datos de Echonest™ se debe tener la canción en formato digital y utilizar las librerías disponibles de forma gratuita para usuarios registrados. Asimismo para poder utilizar estas librerías, en el propio código se debe implementar una *API* a la que se accederá a los métodos que se quiera, mediante la *API KEY*. En esta propuesta se han utilizado las librerías disponibles de Java™. En el siguiente ejemplo se muestra la línea de código necesaria para llevar esto a cabo.

```
EchoNestAPI api = new EchoNestAPI("FILDTEOIK2HBORODV");
```

Ejemplo 3: Creación de una api mediante la *API KEY*[32]

Una vez obtenida la canción en formato digital de la que se pretenden extraer las características, es necesario subirla a los servidores de Echonest™ mediante su método correspondiente, crear un objeto "*Song*" (que es del tipo *JSON*) y obtener el identificador de la canción mediante otro método una vez subida. A partir de aquí se utilizan de forma individual los métodos de la librería para obtener las características de ese objeto y almacenarlas.

Antes de realizar cualquiera de estos dos métodos, en el propio código, se crean tantos objetos del tipo *JSON* como canciones se vayan a obtener, en esta propuesta 50 por lista por año. Más adelante, con los algoritmos mencionados, se dan valores a estos objetos. Finalmente, se utilizan las librerías de *POI HSSF*, que sirven para la creación y manejo de archivos *Excel*™, formato en el que se devuelven todos los datos que se han extraído a través de los servicios que ofrece Echonest™.

Se dan valores celda por celda al fichero en formato *Excel*™ que se crea, organizándolo de la siguiente manera: Cada fila una canción y cada columna

[32] Fuente: Propia

una característica de la misma, de tal forma que al final se obtiene una matriz de 50x12. Debido a requisitos de implementación de esta propuesta así como por cuestiones prácticas, las canciones (filas de características) quedan en el fichero *Excel*™ en el mismo orden en el que aparecen en la lista de canciones que proporciona Billboard™. El orden de las características (columnas), no resulta relevante, de modo que se ha escogido uno al azar, aunque se ha mantenido en todos los ficheros así como en el algoritmo que se describirá a continuación.

Análisis estadístico de los datos recogidos

Una vez realizada la obtención de todas las listas, se tienen entonces 33 hojas *Excel*™, todas de 50x12, que corresponden cada una a una lista de Billboard™ de un año, desde 1980 hasta 2012. Esta parte de la propuesta se ha implementado en el entorno de MATLAB™, haciendo uso sobre todo del paquete de herramientas *Statistics Toolbox*™, que proporciona algoritmos y herramientas para organizar, analizar y modelar datos.

Para comenzar los diferentes análisis que se han realizado y acorde con los objetivos planteados, se etiquetan las canciones (cada fila) de cada lista de la siguiente manera: Las 20 primeras pertenecen a clase '1', es decir, que se consideran exitosas. Las 30 restantes pertenecen a la clase '0', es decir, se consideran no exitosas. Se ha elegido 20 como número de canciones exitosas en una lista debido a que no es una cifra tan restrictiva como 10 (Ej: '*Top 10*'), y es lo suficientemente grande como para proporcionar una cantidad razonable de datos de la clase '1'.

Antes de empezar al análisis se ha eliminado una característica de todas las listas, '*key signature*', debido a que se ha observado que en aproximadamente un 96% de las canciones esta característica toma el mismo valor, 4. Por tanto se ha considerado a priori no relevante para la clasificación, con lo que las matrices de cada lista se quedarían con un tamaño 50x11.

- **Porcentaje de acierto calculado de forma causal utilizando listas de 3 años consecutivos, desde 1980 a 2012**

El primer estudio realizado propone determinar el porcentaje de acierto que se alcanza al clasificar la mitad de una lista de un año (canciones pares), utilizando como datos de entrenamiento la otra mitad de esa lista (canciones impares) y dos listas completas de los dos años anteriores. Se ha escogido como 'ventana' óptima 3 años, ya que mediante diferentes pruebas de ensayo y error se ha determinado que la componente temporal que se busca para la obtención de resultados y el cumplimiento de los objetivos propuestos desaparece al utilizar otras cifras[33].

Para ello se deciden utilizar 3 modelos de métodos de clasificación estadísticos conocidos: Clasificación por KNN o '*K-nearest neighbours*', Clasificador *Naive Bayes* y Clasificación por SVM o Máquinas de Vectores de Soporte, que se implementarán a través de las funciones que ofrece MATLAB™ `ClassificationKNN.fit()`, `NaiveBayes.fit()` y `svmtrain()` respectivamente. El código que se ha utilizado en esta parte de la propuesta calcula los índices de acierto que consiguen estos tres métodos diferentes y escoge el mayor como resultado final.

El algoritmo que se ha implementado recibe 3 listas en *Excel™* como parámetros de entrada. Para comenzar es necesario normalizar todos estos datos, tanto el conjunto de entrenamiento, en este caso una matriz 125x11 (formada a partir de 125 muestras, eso es, dos listas y media) como el conjunto de test, en este caso una matriz 25x11 (25 muestras, media lista). Se procede con una normalización para obtener variables con media cero y varianza uno: Se resta la media y se divide por la desviación típica. Ya que las características se encuentran cada una en una columna, este proceso se ha llevado a cabo por columnas. En el siguiente ejemplo se muestra cómo, donde *X* es el conjunto de valores de característica a normalizar, *mu* la media del conjunto de estas

[33] Ver Gráficas 1 y 2 en el Capítulo 'Validación Experimental'

características, *sigma* su desviación estándar y *Z* los valores de la característica normalizada.

$$Z = \frac{X - \mu}{\sigma} \quad ; \quad Z \sim N(0,1)$$

Ejemplo 4: Normalización de una característica. Obtenido de [1]

Una vez normalizados los datos se procede al modelado de los diferentes clasificadores que se van a utilizar.

Las funciones que implementan los diferentes clasificadores admiten parámetros de entrada que condicionan la manera de llevar a cabo la clasificación. Es por eso que resulta útil elegir estos parámetros de forma adecuada. De los parámetros que ofrecen estas funciones se han decidido modificar los más significativos, siendo estos los que definen en primer lugar el clasificador de forma básica. El resto de parámetros no se han modificado, con lo que MATLAB™ les otorga valores por defecto[34]. Los parámetros que toman valores numéricos se han optimizado mediante un algoritmo de validación cruzada, y los parámetros no numéricos se han establecido de forma manual y son constantes en todas las llamadas a las funciones de los clasificadores.

- *Elección de los parámetros no numéricos*

Tanto en la implementación del algoritmo de validación cruzada como en entrenamiento de los clasificadores, existen parámetros de entrada que se mantienen en ambos modelos, y que se han decidido de manera que optimicen los modelos de los clasificadores que se crean.

En el caso del clasificador *KNN* se ha modificado uno. Este parámetro no numérico es el sistema de medida que se utiliza para computar la distancia entre

[34] Para más información sobre los parámetros de entrada de las funciones que se utilizan ver:
http://www.mathworks.es/es/help/stats/svmtrain.html
http://www.mathworks.es/es/help/stats/classificationknn.fit.html
http://www.mathworks.es/es/help/stats/naivebayes.fit.html

la nueva muestra y sus vecinos. Se ha elegido el valor `'cosine'`, que implementa distancias de coseno. En el siguiente ejemplo se muestra la ecuación para calcular dicha distancia, donde x_s es un vector que contiene los valores de las características de una muestra del conjunto entrenamiento (una canción), y_s es un vector que contiene los valores de las características de una muestra del conjunto de test, y d_{st} es la distancia calculada entre los dos.

$$d_{st} = \left(1 - \frac{x_s y_t'}{\sqrt{(x_s x_s')(y_t y_t')}}\right)$$

Ejemplo 5: Distancia de coseno[35]

Dado que el número de columnas de los datos de entrenamiento es mayor que 10, MATLAB™ establece por defecto como método de búsqueda de vecinos 'búsqueda exhaustiva', es decir, que se computan todas las distancias que existen de la nueva muestra a todas las demás para hallar las menores. Esto implica que MATLAB™ restringe los métodos de medida que se pueden utilizar. De la lista que MATLAB™ ofrece, el utilizar distancias de coseno ofrece una solución sencilla al problema de que normalmente se comparan un número elevado de muestras con la nueva.

En el caso del clasificador *Naive Bayes* se ha modificado el parámetro que define el tipo de distribución que se asume a priori, *P(X|Y)*, y se le ha dado el valor `'kernel'`, que asume que las características existentes son independientes las unas de las otras, y modela una función de densidad para cada una de ellas basándose en los datos de entrenamiento. La elección de este parámetro no requiere unos grandes supuestos, al contrario que en el caso de la distribución normal, y resulta útil en casos donde la distribución de una clase pueda contener múltiples picos o modas.

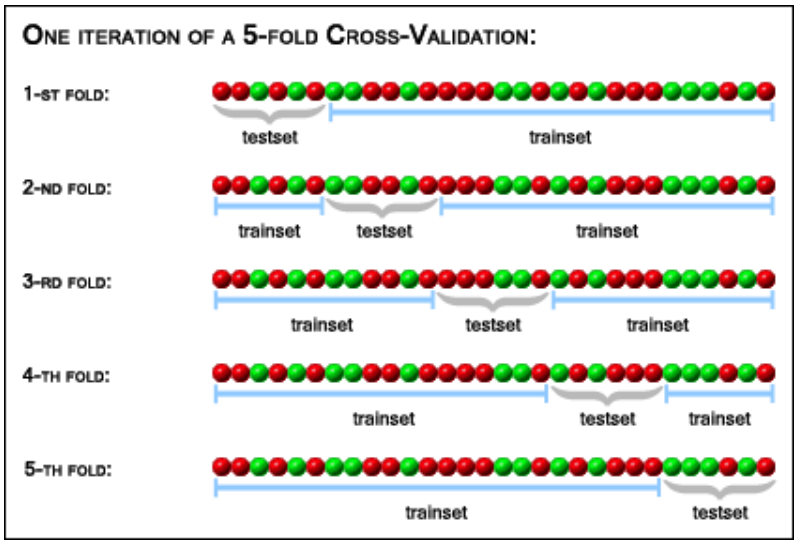
En el caso del clasificador SVM se ha modificado un parámetro. El parámetro no numérico que se ha decidido modificar es la elección del tipo de mapeado de

[35] Obtenido de www.mathworks.es

datos en el espacio kernel. Se ha elegido el valor 'rbf', ya que la geometría del problema que se plantea no es separable.

- *Elección de los parámetros numéricos*

A fin de automatizar el proceso por el cuál se escogen los parámetros que toman valores numéricos de forma justa más adecuados para cada uno de los modelos de los clasificadores, se implementa un algoritmo de validación cruzada de K iteraciones, que desordena los datos de forma aleatoria (para hacer el algoritmo justo, es decir, que los 'folds' que se crean no contengan únicamente canciones del grupo '1' o del grupo '0'), y que posteriormente divide los datos de entrenamiento en 10 'folds'. Se van iterando entonces de forma consecutiva los 'folds' de uno en uno como conjunto de test, y el resto como conjunto de entrenamiento en los clasificadores. En el ejemplo 6 se muestra una iteración en una validación cruzada con 5 *folds,* teniendo los datos de entrenamiento divididos en 2 clases.



Ejemplo 6: Iteración con 5 *folds* en una validación cruzada con un grupo de entrenamiento con dos etiquetas. Obtenido de [2]

Este proceso se repite dando un abanico de valores de forma razonada a los parámetros clave de entrada de los diferentes modelos de los clasificadores: En el caso del clasificador KNN se dan valores al número de vecinos *k*, y en el caso del clasificador SVM el parámetro *sigma* y *c* (parámetro de regularización). Debido a que el clasificador *Naive Bayes* no recibe ningún parámetro de entrada numérico que optimice la clasificación, éste no se ha incluido en el algoritmo de validación cruzada.

Para el clasificador KNN, el parámetro *k* toma valores impares desde 1 a 19: Si el valor es impar, al comparar el clasificador la nueva muestra con ese número de vecinos siempre deberá se deberá decantar por un grupo u otro, es decir, no podrá ser imparcial. Se ha elegido 19 como valor máximo de *k* ya que se ha considerado un número aceptable dados los datos de entrenamiento (125 muestras).

Para el clasificador SVM, el parámetro *sigma* toma valores desde la raíz cuadrada del número de variables por muestra (en esta propuesta, 11 características), doblándose su valor hasta 10 veces. El parámetro *c* toma valores en potencias de 10, desde 10^0 hasta 10^5. Se ha comprobado experimentalmente que estos rangos para los parámetros resultan adecuados para la propuesta.

Finalmente así, en el algoritmo de validación cruzada, comparando las etiquetas que se devuelven en cada clasificación con las reales (en cada iteración), se crea en ambos casos una matriz que contiene los porcentajes de error que corresponden a la utilización de diferentes valores de los parámetros clave. Se escoge entonces el mínimo de dichas matrices, y utilizando los índices de su posición se consiguen obtener los valores de los parámetros por los cuales se dio ese mínimo. De esta manera se han escogido los valores que se introducen en el modelo a la hora de realizar la fase de entrenamiento.

Una vez determinados los parámetros adecuados para los clasificadores, se procede a la creación de los modelos de estos, a partir de los datos de entrenamiento (matriz de 125x11). Se crean los 3 modelos de los 3 clasificadores anteriormente mencionados con sus funciones correspondientes, y a

continuación se procede a evaluar los datos de test (matriz de 25x11). Ya evaluados los datos de test se comparan las etiquetas de clasificación que han devuelto los clasificadores con las reales para obtener el porcentaje de acierto. De los 3 porcentajes de acierto obtenidos se escoge el mayor de ellos como valor final. Se elige esta manera de combinar los clasificadores para aislar nuestro diseño de la elección de una tecnología de clasificación concreta, ya que el clasificador *Naive Bayes* se basa en un modelado generativo probabilístico, el clasificador *SVM* es discriminativo y semiparamétrico no lineal y el clasificador *KNN* es discriminativo no paramétrico.

Este proceso devuelve sólo el porcentaje de acierto de la clasificación de un año. Se procede entonces a ejecutar este código con las 33 listas, para poder ver una evolución a lo largo del tiempo de dichos índices de acierto. Siendo un análisis causal, todas las clasificaciones de cada año dependen de sí mismas y de dos años anteriores. Para la clasificación realizada en los dos primeros años, 1980 y 1981, se han utilizado como excepción menos datos de entrenamiento: Para 1980 la otra mitad de su lista (25 muestras) y para 1981 la lista del año anterior más la otra mitad suya (75 muestras). El proceso de clasificación realizado para éstos dos años es idéntico al anterior mencionado, con la excepción de utilizar menos datos para los conjuntos de entrenamiento respectivos.

- **Análisis no causal de qué características no son relevantes, desde 1980 a 2012**

Este segundo estudio propone realizar un análisis no causal determinando qué dos características por año de las que se utilizan para la clasificación dan los peores resultados, es decir, las que aportan ruido o poca información. De esta manera, analizadas en conjunto, se pueden ver cuáles son las menos relevantes para la clasificación cada década.

A la hora de hallar el porcentaje de acierto de la clasificación se utiliza el mismo método que se ha descrito en el punto anterior, solo que en este caso en lugar de utilizar como datos de entrenamiento los dos años anteriores más la mitad del

año a clasificar, se utilizan el año anterior y el siguiente. De esta manera la clasificación que resulta no es causal.

Se realiza este método a lo largo de los 33 años. Para calcular las características que no son relevantes a posteriori, se implementa el método anterior entrenando y clasificando eliminando 2 características de las tablas de datos. Se iteran las diferentes parejas de características de tal manera que se obtiene una matriz 11x11 que contiene los aciertos que se han dado eliminando dos características diferentes en cada iteración.

Se crea un vector que contiene todos los nombres de las características, en el orden en el que se ejecutan las iteraciones anteriores. De esta manera es posible obtener el máximo de la matriz y con él los índices que indican las características que se ha eliminado en esa iteración, resultando así las que siendo eliminadas producen el mayor porcentaje de acierto, es decir, las que no son relevantes para clasificar.

- **Análisis de la década de los 2000 (2000-2012)**

Después de realizar los análisis anteriores se ha decidido hacer un estudio de las listas que comprenden los años 2000-2012, ya que resulta interesante debido a los resultados obtenidos: Se observa una tendencia creciente en el porcentaje de acierto[36]. Este estudio se centra en averiguar si es posible mejorar la clasificación eliminando dos o más características de cada año de esta década y de buscar los factores que han provocado el éxito en las canciones que han compuesto estos años. Se compone de un análisis a priori de las características de dichas listas y de un cálculo realizado de forma no causal del porcentaje de acierto eliminando las mismas.

[36] Ver Gráficas 3 y 4 en el Capítulo 'Validación Experimental'

- *Análisis a priori de las características*

Para determinar a priori si las características que componen las listas de la década de los años 2000 son relevantes para la clasificación, se ha decidido crear un histograma de cada una de ellas, tanto en la clase '1' (canción exitosa) como en la case '0' (canción no exitosa). Con esto se pretende ver a priori las diferencias que se originan entre las distribuciones que reflejan los histogramas por clases, de tal manera que en un principio se podría determinar qué características ayudan a la clasificación (aquellas cuyos histogramas se diferencien) y cuáles no ayudan (aquellas cuyos histogramas sean similares). Es posible que haya una característica en la que las distribuciones de las clases se solapen mucho, pero que al combinarla con otra característica haga que esas distribuciones se separen. Esto hace que este análisis no se tome como definitivo y se decida profundizar.

Entonces, para este estudio se separan y almacenan las canciones de las diferentes clases en dos matrices. Las matrices entonces se normalizan por columnas (características), del mismo modo que se ha descrito anteriormente: Se resta la media y se divide por la desviación estándar.

Con las funciones `hist()` y `bar()` se crean una serie de histogramas normalizados, uno por característica y clase, es decir, que en total se tienen 22 histogramas, 11 de cada clase, que se ilustran por parejas de forma superpuesta (histogramas de la característica perteneciendo al grupo '1' y '0') en la misma imagen[37].

- *Análisis no causal del cambio en el porcentaje de acierto por eliminación de características de la década de los 2000*

En este análisis se persigue capturar las interacciones entre las características que escaparían al análisis independiente realizado anteriormente, así como los

[37] Ver Gráfica 4 en el Capítulo 'Validación Experimental'

cambios en los aciertos que se consiguen en las clasificaciones si se eliminan diferentes características, concretamente si es posible conseguir un aumento significativo de los aciertos durante esta década. Se realiza en cada año y de forma no causal con una 'ventana' de 3 años, es decir, que una clasificación toma como datos de entrenamiento las listas del año anterior y posterior así como las canciones pares de la lista de ese año (125 muestras de entrenamiento) y como datos de test las canciones impares (25 muestras de test).

En esta parte de la propuesta el análisis estadístico se realiza de forma similar a los análisis anteriores, salvo que sólo se utiliza un algoritmo de clasificación, *KNN*. Esto es debido a que el coste computacional que supone realizar todas las iteraciones requeridas para este análisis incluyendo las de los 3 clasificadores anteriormente mencionados, resulta inviable con los medios informáticos de los que se disponen. Otro motivo de la elección de la clasificación *KNN* es que resulta un algoritmo flexible. Por tanto, el único porcentaje de acierto que se maneja es el que devuelve la clasificación *KNN*.

El algoritmo de eliminación de características utiliza el mismo algoritmo de clasificación que se ha mencionado anteriormente, así como la selección de las características menos relevantes. Este algoritmo se itera de tal manera que a cada iteración se eliminan dos características y se obtiene el acierto que se consigue al clasificar.

Al igual que en el anterior análisis por características de las 3 décadas, se crea un vector que contiene todos los nombres de las características, en el orden en el que se ejecutan las iteraciones anteriores. De esta manera es posible obtener el máximo de la matriz y con él los índices que indican las características que se ha eliminado en esa iteración, resultando así las que producen el mayor porcentaje de acierto, es decir, aquellas que no son relevantes para clasificar.

Este proceso se repite otra vez, eliminando al comienzo las dos características obtenidas anteriormente. Entonces se compara el máximo que ha resultado de tal proceso y se compara con el anterior: Si es mayor o igual quiere decir que las 2 características nuevamente eliminadas no eran relevantes, y por tanto se puede proceder a repetir este proceso. De esta manera se van eliminando

diferentes características en la clasificación en cada año, y observando si el acierto aumenta o no al realizarlo. Las iteraciones se detienen cuando al realizar un análisis eliminando 2 características, el mayor acierto obtenido es menor al que se obtuvo anteriormente, es decir, que se están eliminando características que son relevantes para la clasificación.

Capítulo 4: Validación Experimental

En esta parte del trabajo se procederá a explicar los diferentes resultados que se han obtenido en los análisis descritos en la parte anterior. Todas las gráficas que aquí se muestran han sido calculadas usando el entorno MATLAB™.

- **Porcentaje de acierto calculado de forma causal utilizando listas de 3 años consecutivos desde 1980 hasta 2012**

Antes de comenzar a describir los resultados obtenidos es necesario hacer un pequeño comentario sobre las cifras que estos otorgan. El conjunto de test que se utiliza para todos los análisis es de 25 muestras, que se clasifican en dos grupos, grupo '1' (canciones exitosas) y grupo '0' (canciones no exitosas) siendo la distribución real de estas etiquetas en todos los casos: 10 primeras en el grupo '1' y 15 restantes en el grupo '0'.

A la hora de clasificar, se mide el acierto obtenido comparando las etiquetas de las muestras de test con las etiquetas reales de estas. Con lo cual, podemos decir que al tener 25 muestras de test, 10 de ellas de una clase y 15 de otra, la clasificación más trivial resultaría en un índice de acierto del 60%. Esto es, clasificar todas las muestras como del grupo '0' al ser el más abundante. Por tanto si una clasificación devuelve un índice de acierto inferior al 60%, se considera que el clasificador no ha trabajado correctamente y la clasificación es fallida, ya que la decisión trivial estaría más acertada.

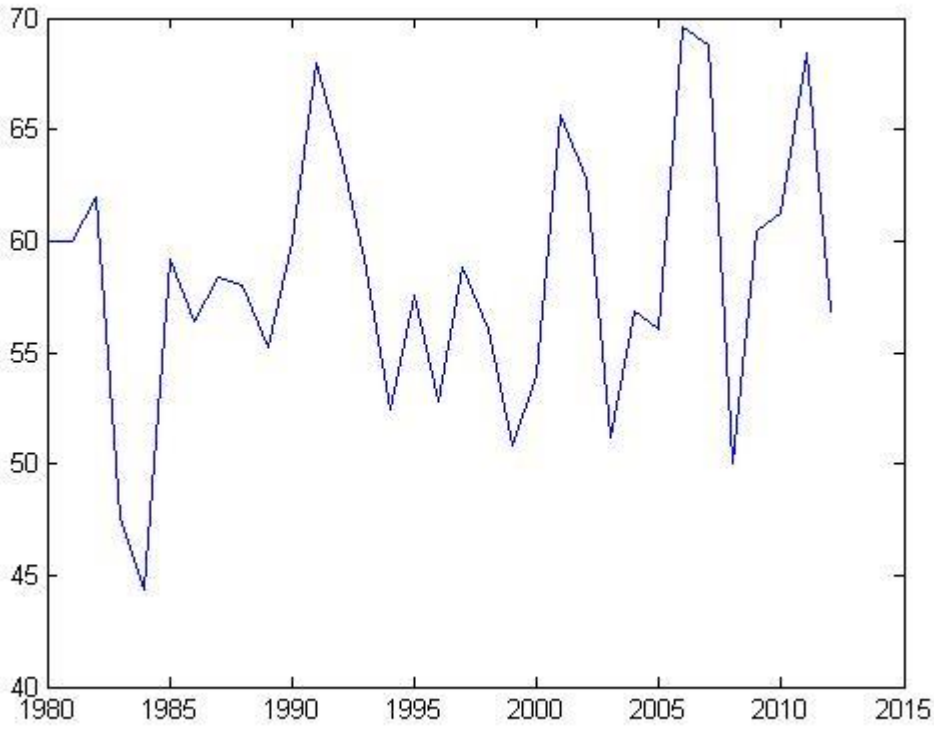
Procediendo de nuevo con el análisis, como se ha descrito anteriormente este calcula el índice de acierto al clasificar media lista de un año usando como datos de entrenamiento las listas de los dos años anteriores y la mitad de ese año, es decir, que se ha usado una 'ventana' de 3 años para la elección de datos de entrenamiento. Esto ha sido así ya que mediante un método de ensayo y error se han realizado experimentos con 'ventanas' de 2 y 5 años (desde la lista del año a

clasificar hacia atrás se entiende, de la misma manera que con 3 años), para comprobar finalmente que estas elecciones eliminan la componente temporal que se busca para la clasificación, esto es, que originan que los datos de entrenamiento se dispersen demasiado, lo que desemboca en que la clasificación no se pueda optimizar.

Dicho de una manera vulgar y desde un punto de vista musical, los gustos musicales son más o menos estables en un periodo de 3 años. 5 años resultan demasiados y 2 años muy pocos si se busca un mínimo de homogeneidad musical en las listas de éxitos.

En las gráficas 1 y 2 se muestran los resultados de esos experimentos:

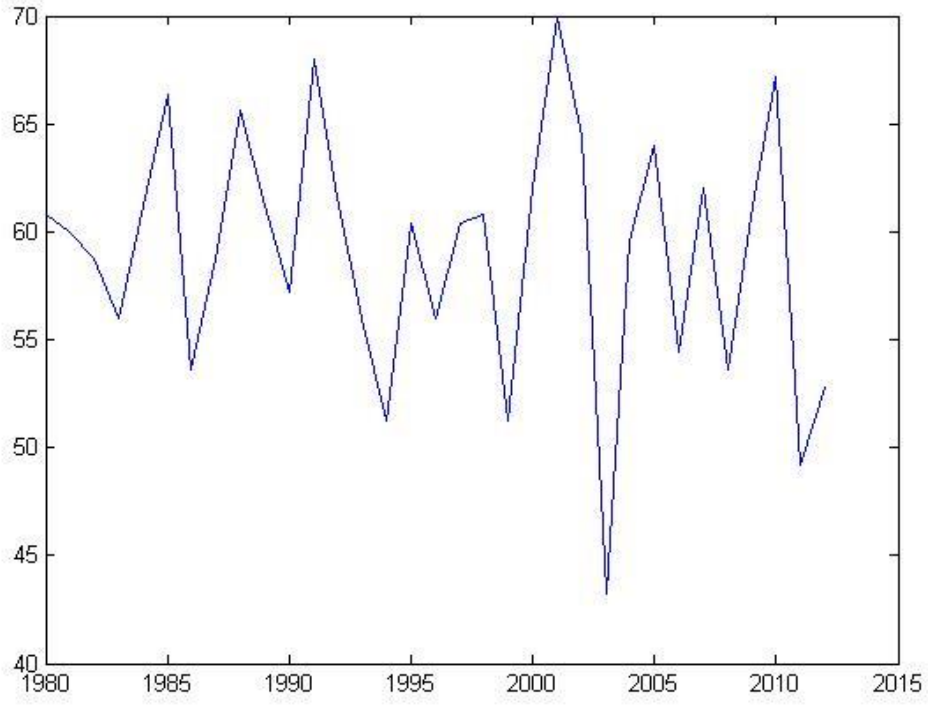
Eje Y: Porcentaje de Acierto; Eje X: Año



Gráfica 1: Porcentaje de acierto utilizando una ventana de 2 años[38]

[38] Fuente: Propia

Eje Y: Porcentaje de Acierto; Eje X: Año



Gráfica 2: Porcentaje de acierto utilizando una ventana de 5 años[39]

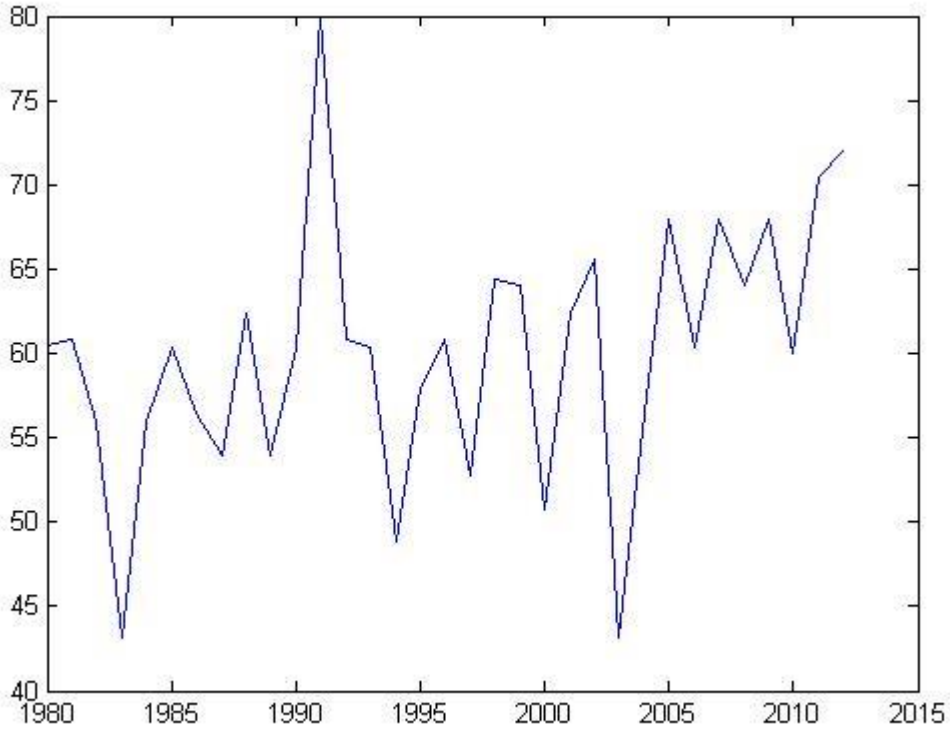
En la primera gráfica se comprueba que la elección de 2 años como 'ventana' para los datos de entrenamiento no ofrece aciertos significativos, la mayoría no pasan del 60%, mínimo establecido para una clasificación aceptable.

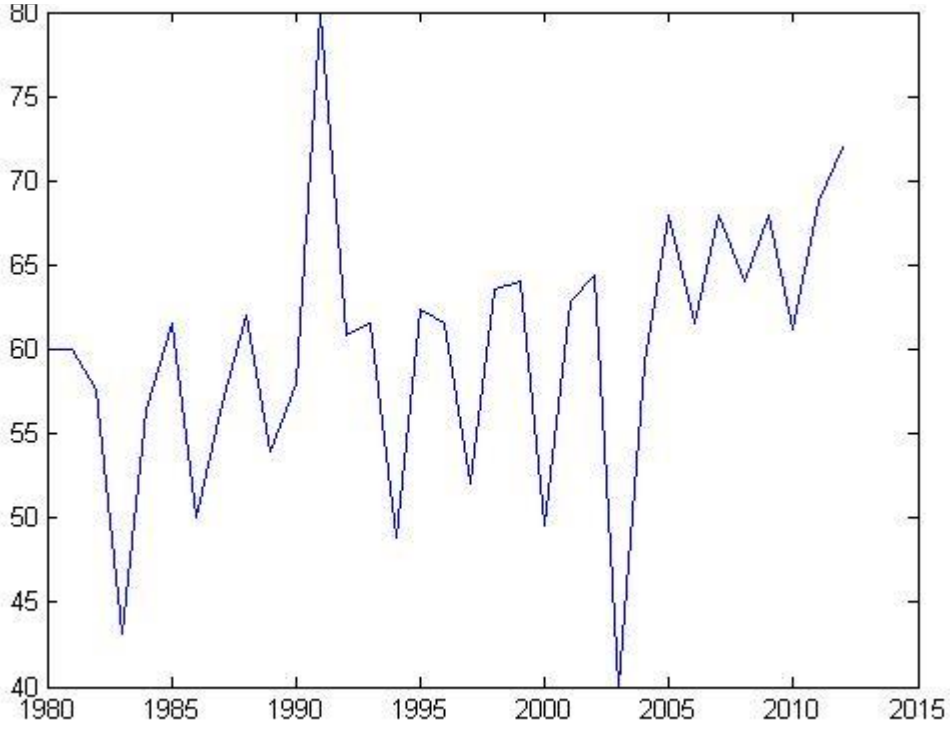
Se puede observar además en las dos gráficas que a pesar de contener unos índices de acierto muy variables, ninguna contiene alguna tendencia en particular. Por esa razón se consideró adecuado el entrenamiento de los diferentes clasificadores con una 'ventana' de 3 años.

A continuación en las gráficas 3 y 4 se muestran los resultados que se han obtenido utilizando 3 años como 'ventana':

[39] Fuente: Propia

Eje Y: Porcentaje de Acierto; Eje X: Año





Gráficas 3 y 4: Porcentaje de acierto utilizando una ventana de 3 años[40]

[40] Fuente: Propia

Se han mostrado dos gráficas pertenecientes a dos simulaciones hechas con una ventana de 3 años para confirmar los resultados, debido a que el algoritmo que automatiza la elección de parámetros puede escoger valores diferentes en cada simulación.

Se puede observar que a pesar de aparecer varios índices inferiores a 60% y existir un desorden aparente como en las gráficas anteriores, existe una tendencia creciente del acierto a partir de aproximadamente el año 1997 y que es perfectamente visible a partir del año 2004.

Por tanto las canciones de las listas a partir de esos años forman grupos relativamente homogéneos que coinciden con la etiquetación que se les ha dado, es decir, que tanto el grupo '1' como el '0' resultan predecibles de manera creciente a medida que se avanza en los años.

- **Análisis no causal de qué características no son relevantes, desde 1980 a 2012**

En este análisis se han determinado qué 2 características resultan las menos relevantes cada año, es decir, aquellas que eliminándolas de las matrices de datos proporcionan los índices de acierto más altos. Este análisis se ha determinado desde 1980 a 2012, y resulta un primer acercamiento a la relevancia de las características que aquí se manejan, aunque más adelante se realiza un estudio similar que elimina todas las características que puedan causar una mala clasificación en la década de los 2000, por tanto es posible que los resultados estén sujetos a cambio en este periodo, cabiendo la posibilidad que existan variables que introduzcan ruido en la clasificación de tal manera que interfieran con las variables que sí resultan útiles para la clasificación y con las conclusiones que se les han aplicado.

El resultado que se ha obtenido de esta simulación no es una gráfica sino una serie de cadenas de texto separadas por décadas que describen las diferentes características eliminadas. Por tanto se va a proceder a ordenarlas y mostrarlas de tal manera que se pueda ver con claridad dicho resultado:

Año / Caract.	80s	90s	00s
Danceability	**IIII**	**III**	**~~IIII~~**
Duration	**III**	**IIII**	**II**
Energy	**I**	**I**	**III**
Key	**II**	**III**	**IIII**
Loudness	**I**		**II**
Mode	**I**		**III**
Speechiness	**I**		
Acousticness	**IIII**	**II**	**I**
Liveness	**I**	**II**	**I**
Tempo	**I**	**III**	**II**
Valence	**I**	**I**	**III**

Tabla 1: Número de veces que las características resultan no relevantes por década

En la tabla superior se muestran las veces que el algoritmo ha determinado que una característica era no relevante para la clasificación. Se han distribuido por décadas para poder hacer un análisis general de todos los años y así tener una perspectiva global de lo que ocurre con la clasificación en los diferentes periodos de tiempo.

Observando la tabla se puede determinar que los parámetros *Danceability*, *Duration* y *Key* aparecen más veces y por tanto resultan menos relevantes para la clasificación en las 3 décadas, lo que quiere decir que en general la componente que hace diferenciar una canción que se considera exitosa de una que no, no reside en estas características. Se observa también, que hay ciertas características que aparecen más o menos veces en diferentes décadas, y que hay características que apenas aparecen.

Se puede observar que la característica *Speechiness* es aquella que aparece un menor número de veces. Ya que este parámetro valora la cantidad de palabra

hablada en una canción[41], se puede llegar a la conclusión de que en las tres décadas la palabra hablada juega un papel crucial en la clasificación. Algo similar ocurre para la característica *Loudness,* que mide el volumen promedio en decibelios[42].

Resulta interesante observar como ciertas características aparecen más o menos veces al aumentar el año. Dos ejemplos claros que se pueden observar son las características *Acousticness* y *Key*:

El número de veces que la característica *Acousticness* aparece, desciende con el año. Esto quiere decir que resulta ser una característica más y más relevante al clasificar a partir de 1980.

La característica *Key* determina la nota tónica principal de una canción, su clave, y se puede observar que el número de veces que se considera no relevante aumenta con el año.

Las características *Valence* y *Energy* aumentan de forma similar a la mencionada anteriormente, con lo que se pueden aplicar las mismas deducciones: Que su relevancia en las clasificaciones decrece con el año.

- **Análisis de la década de los 2000 (2000-2012)**

En este análisis se procede a realizar un estudio centrándose en la década de los años 2000, debido a que se ha comprobado que hay un porcentaje de acierto creciente en esta década, lo que ha llamado la atención.

- *Análisis a priori de las características*

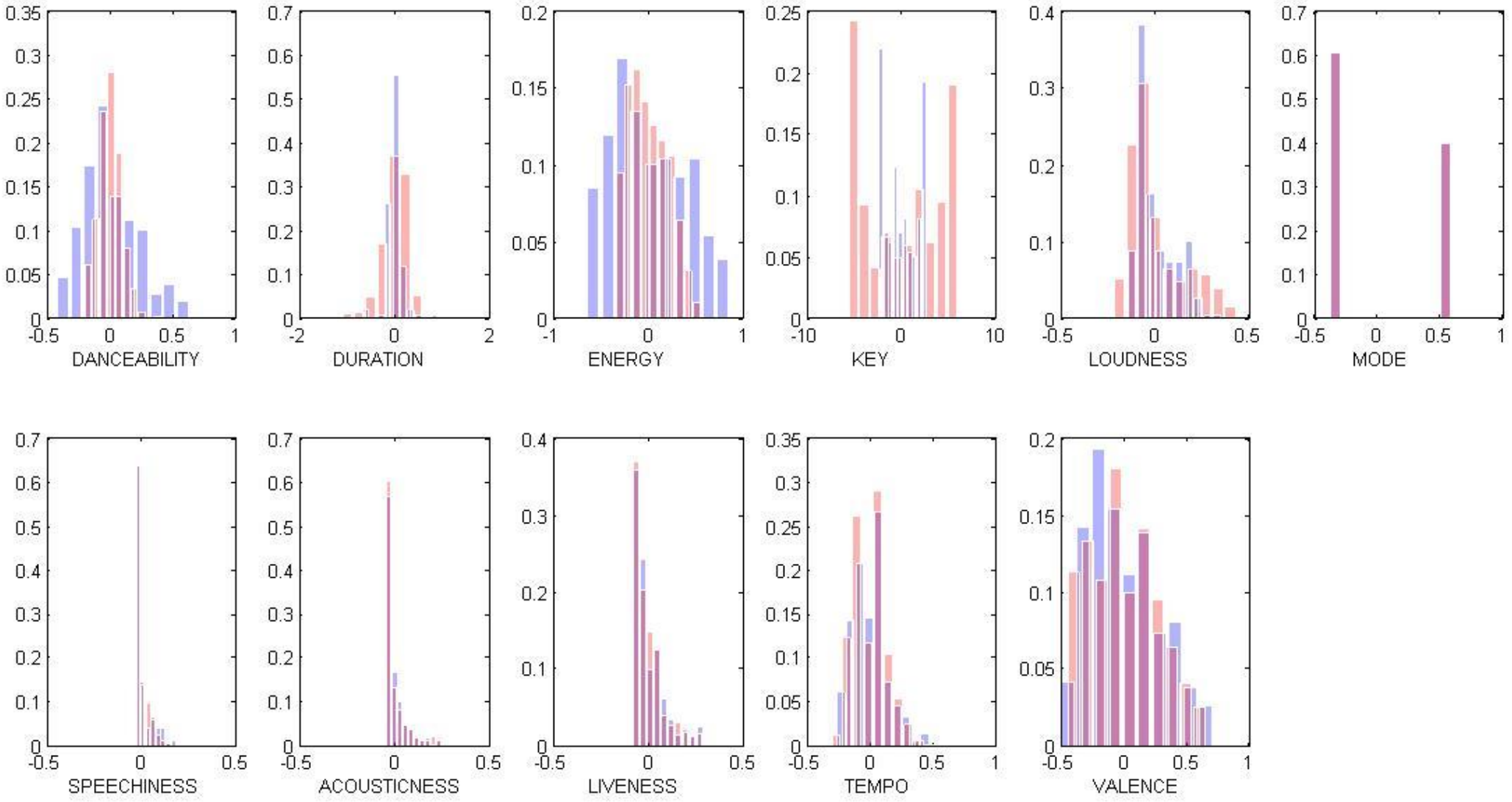
Como se ha comentado anteriormente, se ha realizado un estudio a priori de las características en esta década. A continuación en la gráfica 5 se muestran los histogramas normalizados resultantes, categorizados por característica y clase.

[41] Ver Capitulo 2 'Estado del Arte'
[42] Ver Capitulo 2 'Estado del Arte'

Cada pareja de histogramas pertenece a una característica. Los histogramas en color azul representan a la característica correspondiente en la clase '1' (canción exitosa) y los histogramas en rojo a la misma característica.

Para facilitar la vista de tales histogramas se han transparentado en cierto grado los mismos, de tal manera que las barras en color morado son las superposiciones de dos barras de las dos clases.

Gráfica 5: Histogramas normalizados de las diferentes características agrupadas por clases



Se deduce entonces, que de este análisis a priori las características que contribuirían más a la clasificación serían aquellas que contienen histogramas diferentes (recordar que este estudio es a priori y toma las características de forma independiente, sin analizar la interacción entre ellas[43]): *Danceability, Energy, Key, Loudness* y *Speechiness*.

- *Análisis no causal del aumento en el porcentaje de acierto por eliminación de características de la década de los 2000*

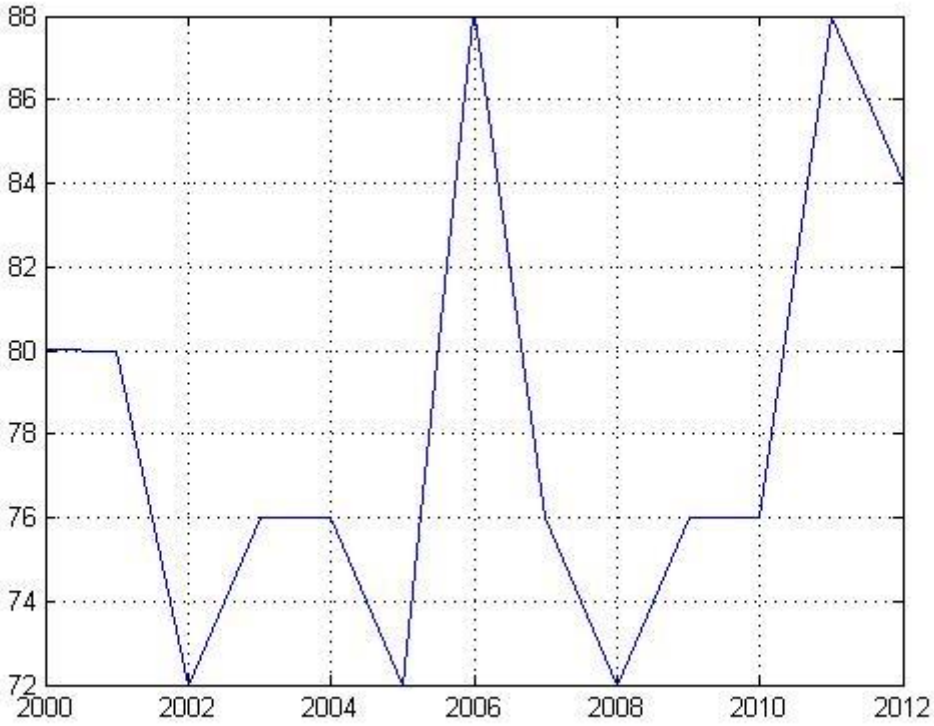
En este análisis se procede a observar el aumento del porcentaje de acierto obtenido de forma no causal en la década de los años 2000 si se procede a eliminar diferentes características que aportan ruido a la clasificación. Para ello se ha empleado el mismo algoritmo de cálculo del acierto en la clasificación que en los anteriores análisis, con la diferencia de que el porcentaje de acierto que se obtiene proviene únicamente de la clasificación utilizando un clasificador *KNN*.

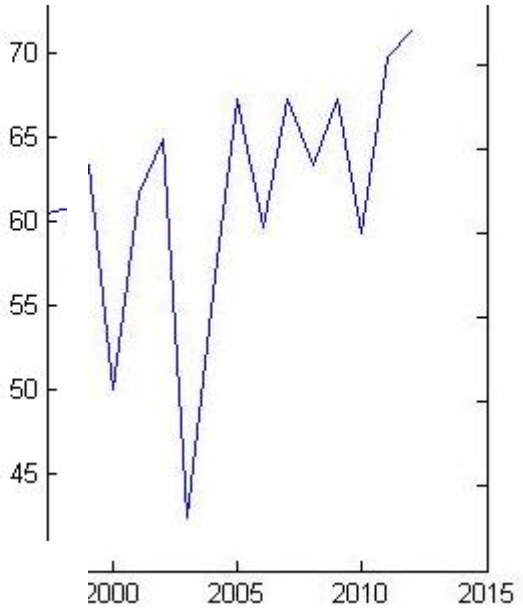
El algoritmo de eliminación de características funciona de forma iterativa, eliminando de las matrices de datos dos características a cada iteración, y comparando el mayor porcentaje de acierto obtenido con el que se obtuvo en la anterior iteración (eliminación de una pareja de características). De tal manera que el algoritmo se detiene al intentar eliminar una pareja de características que producen un acierto menor que el mayor obtenido en la anterior iteración.

Los resultados que se han obtenido son, por un lado la gráfica que contiene dichos porcentajes de acierto, y por otro las características que se han eliminado de forma iterativa en cada año. A continuación en la gráfica 5 se muestran dichos resultados. En la gráfica 6 se muestra una parte de la gráfica 3 para facilitar la comparación de las dos gráficas.

[43] Ver Capítulo 3

Eje Y: Porcentaje de Acierto; Eje X: Año





Gráficas 5 y 6: Porcentaje de acierto eliminando de forma iterativa diferentes características cada año en la década de los 2000 y parte de la gráfica 3 para poder comparar resultados[44]

[44] Fuente: Propia

Esta gráfica muestra el porcentaje de acierto que se consigue al eliminar características que aportan ruido o hacen disminuir dicho porcentaje. De entrada, si se comparan los porcentajes mínimos y máximos a los que llegan la gráfica anterior y la gráfica calculada con 'ventana' de 3 años en el primer punto de este apartado, se observa que de forma clara existen características que aportan un claro ruido a la clasificación: En la gráfica del primer punto del apartado 'resultados', en la década de los 2000 se llega a un máximo del 72% de acierto y un mínimo del 43%, mientras que en la gráfica anterior se obtienen un máximo del 88% de acierto un mínimo del 72%. Se deduce entonces que la clasificación resulta muy exitosa al eliminar ciertas características.

Aunque en esta gráfica los porcentajes de acierto varían como máximo en un 16%, siguen siendo unos índices muy altos, lo cual indica que las clases de clasificación grupo '1' (canción exitosa) y grupo '0' (canción no exitosa) se vuelven bien diferenciables si se eliminan de la clasificación características específicas, es decir, que las listas de éxitos en esta década se vuelven altamente predecibles.

En cuanto a las características eliminadas obtenidas del algoritmo implementado, se mostrarán a continuación ordenadas en dos tablas para facilitar su lectura. La primera tabla contiene un código de colores que explica el orden en el que fueron eliminadas dichas características, ya que esto resulta útil para conocer el grado de la relevancia de las mismas (a pesar de ser en conjunto todas ellas no relevantes), y la segunda contiene los porcentajes del número de veces que estas aparecen. El código de color es el siguiente:

- Negro: Característica que se encuentra en el 1er par de eliminadas
- Rojo: Característica que se encuentra en el 2do par de eliminadas
- Amarillo: Característica que se encuentra en el 3er par de eliminadas

Año / Caract.	'00	'01	'02	'03	'04	'05	'06	'07	'08	'09	'10	'11	'12
Danceability													
Duration													
Energy													
Key													
Loudness													
Mode													
Speechiness													
Acousticness													
Liveness													
Tempo													
Valence													

Tabla 2: Características eliminadas de forma iterativa por años en la década de los 2000, coloreadas según su orden de eliminación

Par eliminado/Caract.	1er par	2do par	3er par
Danceability	**23%**	**23%**	**7%**
Duration	**23%**	**7%**	**7%**
Energy	**7%**	**7%**	**7%**
Key	**23%**	**-**	**7%**
Loudness	**15%**	**38%**	**-**
Mode	**7%**	**7%**	**7%**
Speechiness	**7%**	**23%**	**-**
Acousticness	**38%**	**23%**	**7%**
Liveness	**23%**	**15%**	**7%**
Tempo	**7%**	**23%**	**-**
Valence	**23%**	**7%**	**-**

Tabla 3: Porcentajes calculados a partir del número de veces que han aparecido dichas características en la tabla anterior en total

Resulta interesante comparar los resultados que ofrecen las tablas anteriores con el estudio de las características a priori que se ha realizado, para comprobar si las predicciones a priori que se establecieron coinciden con lo obtenido. En el estudio a priori se determinó observando los histogramas que se obtuvieron, las características que aportaban más información a la clasificación eran: *Danceability, Energy, Key, Loudness* y *Speechiness.*

Si se observa la tabla anterior, concretamente la primera columna, aquella que establece el porcentaje del número de veces que esa característica se ha eliminado en la primera iteración del algoritmo (es decir, que aportan mucho ruido), se puede ver que los porcentajes más bajos (características que resultan más relevantes) los adquieren las características: *Energy, Mode, Speechiness* y *Tempo*. El análisis a priori entonces sólo ha coincidido con dos de las características que el algoritmo ha considerado como más relevantes al clasificar, *Energy* y *Speechiness*. Esto quiere decir que estas dos características aportan información a la clasificación de forma individual.

De esta comparación se deduce que existen interacciones entre las características que aportan información a la clasificación, algo que de forma individual no ocurriría, por tanto es necesario que para que la clasificación tenga éxito, se deban analizar dichas interacciones y observar los porcentajes de acierto que se obtienen (utilizando el algoritmo implementado que se ha descrito anteriormente).

Observando individualmente la tabla 3 se tiene que las características *Energy* y *Mode* obtienen los menores porcentajes en los tres pares de características eliminadas con lo que se consideran características altamente relevantes para la clasificación en esta década. Del mismo modo, aunque considerándose menos relevante se encontrarían las características *Key y Valence*: *Key* no aparece en la tabla en la segunda columna, con lo que el algoritmo ha determinado que aporta información en la clasificación, aunque en menor grado (debido a que en la primera iteración ha aparecido varias veces). *Valence* aparece un número reducido de veces en la segunda iteración, y ninguna en la tercera, con lo que del mismo modo se ha considerado relevante en un menor grado.

Se puede observar además qué características se consideran las menos relevantes: En el primer par de características eliminadas (primera columna) se obtiene que es la característica *Acousticness* aquella que obtiene un porcentaje más alto (es escogida un mayor número de veces dentro de la primera iteración), y en el segundo par de eliminadas (segunda columna) se obtiene como característica que más aparece *Loudness*. Además de tener los porcentajes más altos en dichas columnas, obtienen además un porcentaje relativamente algo en los demás pares eliminados (exceptuando en el tercero). Se concluye por tanto que estas dos características añaden ruido y resultan las menos relevantes para la clasificación.

Aunque existen diferentes características que obtienen diferentes porcentajes a lo largo de la tabla, resulta interesante fijarse en estos mínimos y máximos, ya que aportan una gran información sobre la clasificación realizada y ofrecen una buena perspectiva.

Tómese nota de que, el análisis por características efectuado sobre las 3 décadas concluía que para la década de los 2000 el parámetro *Acousticness* resultaba relevante, mientras que en el análisis descrito anteriormente se concluye lo contrario, y que en general y exceptuando las características *Key* y *Speechiness,* las conclusiones en ambos estudios por características no coinciden.

Sin embargo hay que tener en cuenta que este último estudio se ha realizado haciendo uso tan sólo de un algoritmo de clasificación, mientras que en el primero se han usado 3. Además los estudios utilizan conjuntos de datos de entrenamiento diferentes, al ser estudios causales y no causales respectivamente. Es por eso que los resultados de los diferentes estudios son interpretados de manera individual, y por tanto aportan conclusiones independientes.

Capítulo 5: Conclusiones

En esta propuesta se ha tratado de buscar mediante un análisis de audio de las diferentes canciones que componen las listas de éxitos mundiales entre 1980 y 2012, un patrón que las hiciese predecibles, y de ser así, averiguar cuáles han sido los factores que han intervenido en dicho patrón de éxito. Para el análisis de dichas canciones se han determinado dos grupos, que se han usado posteriormente para la clasificación: Canciones exitosas, aquellas que se encuentran en los 20 primeros puestos de la lista, y canciones no exitosas, aquellas que se encuentran en los 30 restantes.[45]

Centrándose en la realización de la propuesta (estudio y análisis de los datos obtenidos a través de los servicios de Echonest™) y resumiéndola, ésta ha constado de 3 análisis:

1. Porcentaje de acierto calculado de forma causal utilizando listas de 3 años consecutivos, desde 1980 a 2012

2. Análisis no causal de qué características no son relevantes, desde 1980 a 2012

3. Análisis de la década de los 2000 (2000-2012)

    - Análisis a priori de las características
    - Análisis no causal del aumento en el porcentaje de acierto por eliminación de características de la década de los 2000

Los resultados del primer análisis mostraban el acierto obtenido al realizar la clasificación mediante una gráfica (representando en cada año el mayor de los tres porcentajes de acierto calculados[46]), en la que se observaba claramente

[45] Detalles en Capítulo 3
[46] Detalles en Capítulo 3

cómo esta contenía una tendencia creciente a partir aproximadamente del año 1997. Este análisis se ha realizado de forma causal, es decir, que las clases de los datos de entrenamiento en un año (canción exitosa/canción no exitosa) han sido predichas en función de los datos de entrenamiento que pertenecen a ese mismo año y a años anteriores. Por tanto, observando esa tendencia en la década de los años 2000, se puede concluir que para esta década la predicción sobre si una canción aparecerá en las 20 primeras posiciones en una lista 'top 100' es posible, con un porcentaje medio de acierto del 64% y con un mínimo del 43% y un máximo del 72%.[47]. Además, la pequeña tendencia creciente en el acierto nos indica que, en esta década, el éxito (pertenencia a los 20 primeros puestos de una lista de éxitos) se vuelve más predecible a medida que pasan los años.

Para las otras dos décadas los resultados, aunque a veces positivos, no contienen tendencias aparentes ni presentan ningún tipo de estabilidad, con lo cual se concluye que la clasificación no es siempre fiable, y que por tanto la predicción no es posible, al menos, habiendo realizado el análisis descrito.

Utilizando las mismas premisas y los mismos algoritmos de clasificación que en el primer análisis, y en el mismo periodo temporal, se han obtenido como resultados del segundo análisis dos características por año que se han considerado no relevantes, todas ellas ordenadas posteriormente en una tabla para una visualización fácil[48].

Se observa que, independientemente de la década y en general, la cantidad de voz hablada y el volumen resultan factores condicionantes al predecir el éxito de una canción, mientras que la 'bailabilidad' y la duración no.

Respecto a la evolución de ciertas características a través de los años se puede especular que:

- La clave resulta muy útil para la predicción en la década de los 80s, donde se concluye que los éxitos estaban compuestos en claves similares,

[47] Según la validación experimental realizada
[48] Detalles en Capítulo 3

mientras que en los años 2000 ocurre al contrario, las claves en las que las canciones están compuestas se encuentran muy dispersas.

- La cantidad de motivos acústicos resulta poco útil para la predicción en los 80, mientras que al ascender la década resulta más relevante. Esto puede ser debido a la creciente emergencia de la música electrónica en las listas de éxitos.

El tercer análisis ha buscado en la década de los años 2000 los factores que han provocado el éxito en las canciones que han compuesto estos años. Este análisis se ha realizado de una forma no causal, es decir, sin utilizar de forma exclusiva como datos de enteramiento años anteriores. Además se ha procedido a utilizar sólo un algoritmo de clasificación, *KNN*. Como resultados se han obtenido una serie de características por año que se han considerado no relevantes (solo que en este caso pueden ser más de dos) y una gráfica mostrando los porcentajes de resultado que se han obtenido al eliminar estas características de los datos. Las características obtenidas se han mostrado en unas tablas para mejorar su visualización.

Observando la gráfica obtenida se ve que existen interacciones entre características que aportan ruido a la clasificación. Teniendo cifras que oscilan entre un 72% y un 88% de acierto[49] se concluye que los dos grupos de canciones, exitosas y no exitosas, están bien definidos, y que por tanto existe un patrón que determina el éxito en ellas. Además, el hecho de eliminar características aumenta de forma significativa el acierto en la clasificación.

Respecto a las características obtenidas, observando los resultados se concluye que la cantidad de palabra hablada, la modalidad de una canción (que sea en un modo mayor o menor) y la energía son factores que definen el éxito en esta década, mientras que la cantidad de motivos acústicos y el volumen promedio no. Especulando, esto puede ser debido a que toda la música popular se produce de forma electrónica y la reciente moda de producir éxitos (de forma similar) incluyendo colaboraciones con raperos de fama mundial. Además, y en relación

[49] Según la validación experimental realizada

a la modalidad y la energía, es un hecho que todos estos éxitos suelen estar compuestos en modalidades similares, por lo general en modos mayores. Esto tiene una correlación directa con la energía ya que canciones electrónicas con modos mayores suelen ser muy energéticas[50] por lo general.

Finalmente, por una parte, se ha determinado que el éxito, es decir, la pertenencia a los 20 primeros puestos en una lista de éxitos, sí es predecible hasta cierto punto en la primera década de los años 2000 y por extensión hoy día. Se ha determinado además que la cantidad de palabra hablada y el volumen promedio son las características que más influyen a la hora de realizar dicha predicción, y que la 'bailabilidad' y la duración son las que menos.

Y por otra parte, al analizar los años 2000 de forma individual, se ha determinado que para esta década en particular las características que mejor definen a las canciones exitosas son la modalidad, la cantidad de palabra hablada y la energía, y que las que peor las definen son la cantidad de motivos acústicos y el volumen promedio.

Habiendo obtenido estos resultados se puede concluir que se han llevado a cabo los objetivos propuestos y se han contestado a las preguntas principales que motivaban esta propuesta.

[50] Ver descripción de características en Capítulo 2

Capítulo 6: Presupuesto

A continuación se detallará el coste que supondría realizar esta propuesta en un entorno profesional.

Coste de personal:

Personal	Número	Coste/hora[51]	Horas de trabajo[52]	Coste total
Ingeniero	**1**	**13.58 €/hora**	**552**	**7496 €**

Listado del material utilizado:

- Ordenador portátil "*Samsung NP-RV511-S04ES*" – ***500 €***
- Licencia individual para uso comercial de MATLAB™[53] - **2000 €**
- Software Eclipse™ - **0 €**
- Librerías varias en Java™[54] - **0 €**
- Acceso a Echonest™ y obtención de *API KEY* - **0 €**
- 16 Canciones descargadas vía ITunes™[55] - **16 €**

[51] Calculado a partir de una estimación del salario anual de un ingeniero, 30.000 €, y el número de días trabajados estimados por año, 280

[52] $23\ x\ 4\ x\ 6\ [dias\ laborables\ x\ meses\ x\ horas\ diarias] = 552\ horas\ trabajadas$

[53] Renovable anualmente

[54] Detalles en Capítulo 2

[55] Aquellas que no se encontraban Echonest™ y ha sido necesario obtenerlas en formato digital

Coste de material:

Material	Unidades	Coste amortización anual/Unidad	Horas anuales[56]	Coste total[57]
Pc	**1**	**250 €**	**2240**	**61.5 €**
Licencia MATLAB™	**1**	**2000 €**	**2240**	**492 €**
Eclipse™	**1**	**0 €**	**2240**	**0 €**
Librerías Java ™	**1**	**0 €**	**2240**	**0 €**
API KEY	**1**	**0 €**	**2240**	**0 €**
Canciones	**16**	**1 €**	**2240**	**16 €**

Resumen de presupuesto

Personal – **7496 €**

Coste de material – **569.5 €**

Coste de material + I.V.A. (21%) – **685.7 €**

TOTAL COSTES – 8181.7 €

Gastos Generales (4%) – **327 €**

Beneficio Industrial (7%) – **573 €**

TOTAL – 9081.7 €

[56] 280 *x* 8 [*dias laborables anuales x horas diarias*]

[57] $\frac{X}{Y} * Coste\ amortización\ anual, donde\ X\ son\ horas\ de\ proyecto\ e\ Y\ son\ horas\ anuales\ de\ trabajo$

Referencias

[1] Juan, A., Sedano M. y Vila A. Distribución Normal. *Proyecto E-Math*. Universitat Oberta de Catalunya

[2] Joanneum, F.H. (2005-2006) Cross-Validation Explained, *Institute for Genomics and Bioinformatics*

[3] Holmes, C. C., Adams, N. M. (2002) A Probabilistic Nearest Neighbour Method for Statistical Pattern Recognition. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*

[4] Moujahid, A., Inza I. y Larrañaga P. Clasificadores KNN. *Departamento de Ciencias de la Computación e Inteligencia Artificial*. Universidad del País Vasco (Euskal Herriko Unibertsitatea)

[5] Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*

[6] Webb, G. I., J. Boughton, y Z. Wang (2005). Not so Naive Bayes: Aggregating One-Dependence Estimators. *Springer Science, Machine Learning, 58,* 5-24

[7] Parrado-Hernandez, E., Gómez-Verdejo, V. y Lázado-Gredilla, M. (2012) Low cost model selection for SVMs using local features. *Engineering Applications of Artificial Intelligence, 25,* 1203-1211

[8] Dasarathy, B.V. (1991) Nearest Neighbour (NN) Norms: NN Pattern Recognition Techniques. *IEEE Computer Society Press*